

## 明 細 書

## ソフトウェアの生産方法

## 技術分野

- [0001] 本発明は、業務用ソフトウェアやゲーム用ソフトウェア、その他あらゆる分野のソフトウェアに適用されるソフトウェアの生産方法、処理装置、開発装置、開発支援装置、管理装置、処理方法及び該ソフトウェアが記録される記録媒体に関する。

## 背景技術

- [0002] どれだけOSやパッケージが良く作られても、それだけでは利用者の要求を満たすことができない。ソフトにはユーザーが自前で開発しなければならない論理部分が必ず存在するからである。重要なことはその部分についてソフトの生産技術が適用されなければならないことである。即ち、生産技術とはソフトの短期開発、保守課題からの解放、上流仕様とプログラム仕様を関数的あるいは理論的に接続する方法、プログラムの機能品質を保証する等の効果が発揮されるものでなければならない。

- [0003] 従来方法は要件機能を抽出してソフト化する機能分割法であるが、機能分割法からは付加的なロジックが派生することが殆どの場で理解されていない。そして、この思考方法が基本的な欠陥であり、すべてのソフト課題はここから生じていると言っても過言でない。

非特許文献1:根来 文生著、「要求定義とソース・コードの間の関係構築のための公理法則に関する研究」、知識ベース・システムについての国際会報、エルスビア・サイエンス出版、2003年秋

非特許文献2:根来 文生著、「Lyeeの仮説的世界」、『ソフトウェア方法論、ツール及び技術における新潮流』予稿集84巻、アイオーエス出版、p. 3-22、2002年

非特許文献3:根来 文生著、「決定論的方法によるソフトウェアの決定のための方法論」、『情報技術及び情報ネットに関する国際会議』予稿集(ICII2001)、会議室D、p. 124-129、2001年

非特許文献4:根来 文生著、「ソースコード生成のための意識操作化論」、システムクス、サイバネティクス及びインフォマティクスについての世界多極会議会報予稿集(

SCI2001 )、ISBN980-07-7554-4、第14巻、コンピュータ科学及びエンジニアリング:第2章、p. 496-503、2001年

非特許文献5:アイ・エイ・ハミド、根来 文生著、「21世紀のソフトウェア実行方法の革新」、『情報技術及び情報ネットに関する国際会議』予稿集(ICH2001)、会議室D、p. 487-489及びp. 235-245、2001年

非特許文献6:根来 文生他著、「意思工学への提言」、インターネット上の電子的商取引、科学及び教育のための基盤の進歩に関する国際会議予稿集(SSGRR2001)、第14回、ISBN88-85280-61-7、CD-ROM、2001年

非特許文献7:根来 文生著、「ソフトウェアの意思表示のための述語構造」、ACISソフトウェア工学、人工知能、ネットワーキング及び並行/分散コンピューティングに関する第2回国際会議予稿集(SNPD'01)、pp985 - 992、2000年

非特許文献8:根来 文生著、「Lyee ソフトウェアの原理」、データベース及び情報システムの進歩に関する第5回東欧会議予稿集(ADBIS2001)、第2巻、個人指導・専門家通信及びレポート、pp7 - 18、2001年

非特許文献9:根来 文生著、「Lyee ソフトウェアの原理」、21世紀における情報社会についての国際会議会報(IS2000)、pp441 - 446、2000年

非特許文献10:イー・ダブリュ・ダイクストラ他著、「構造化プログラミング」、科学出版社、1975年

非特許文献11:ジェイ・ディー・ウォーニア、ビー・エム・フラナガン著、「プログラムの同期化」第1巻「プログラム構造」、第2巻「その実践」、オーガニゼーション版、1971年、1972年

非特許文献12:エフ・ハームセン、エス・ブリンクケンパー著、「CASE環境におけるメソッド・ベース・マネジメント・システムの設計と実践」、第2回APSEC会議予稿集、IEEEコンピュータ社会出版、1995年

非特許文献13:エイ・アール・ガルシアディエゴ(ダントン)著、「バートランド・ラッセルと“パラドックス”集合理論の創設者達」、バーゼル、ビルクホイザ出版、1992年

非特許文献14:ピー・フラスコラ著、「ヴィットゲンシュタインの数学哲学」、ロンドン、1994年

非特許文献15:アール・エイチ・エム・エルヴェス訳、「理解の改良について」、「倫理」ラテン原語からの(1883年)翻訳文、ニューヨーク、ドーヴァ出版、1951年、ISBN0 486 20250 X

## 発明の開示

### 発明が解決しようとする課題

- [0004] 上述したように、従来方法は要件機能を抽出してソフト化する機能分割法であるが、機能分割法からは付加的なロジックが派生することが殆どの場で理解されていない。そして、この思考方法が基本的な欠陥であり、すべてのソフト課題はここから生じていると言っても過言でない。この問題を基本的に改善するにはこの思考方法に替わる新しい概念、即ち、要件事象を一義化する理論が必要である。
- [0005] 本発明者は、かかる理論を案出し、これをLyee(GOVERNMENTAL METHODOLOGY for SOFTWARE PROVIDENCE)と命名した。本理論は意味論についての新たな取り扱い方の基準を与えるものである。
- [0006] ソフトの生産保守性の非効率性及び開発期間の不安定性はソフト構築の手段をソフトの特質を無視する機械機能的なパッケージ化に向かわせている。このパッケージ自体もまた一義性がない点で非正解型である。結果的に利用者の利用環境は次第に信頼性、可用性、適用性(RAS)機能の局面で悪化の一途をたどっている。この時期必要なことは生産の為の基盤整備である。
- [0007] この分野においては、生産技術の問題に手をつけずにいながら、即ち、プログラムの単位的な処理機能を品質的に管理する方法を持たず、上流だけで良質のソフトが決定できると主張するのである。異常と言わねばならない所以である。ソフトの生産方法は上流下流に分けて行われるような性質のものではなく、理論的に捉える所から始めなければならない。Lyee理論は正にこれを満たすものである。
- [0008] 即ち、本発明の目的は以下の目的を達成するソフトウェアの生産方法、処理装置、開発装置、開発支援装置、管理装置、処理方法及び該ソフトウェアが記録される記録媒体を提供することである。
- (1) ソフト開発者及び保守者の生産能力を向上させること。
  - (2) 上流情報と下流情報とを理論的に接続させること。

(3) 単に業務ソフトの開発に限らず、OS、ミドルソフト、ゲームソフト等幅広いソフト分野の開発に適用できるプログラム構造を提供すること。

(4) 上流情報を最小化しソースプログラムを生成すること。

### 課題を解決するための手段

[0009] かかる課題を解決するために本発明は、開発対象のソフトウェアに関する要件を、要件をとらえる最小単位である主語＝[名詞, 狭義の属性, 広義の属性]で定義する第1のステップと、名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉える第2のステップと、前記第1のステップで定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム)：

L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E

を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで前記ベクトルをサブルーチン・プログラムとして形成する第3のステップと、前記サブルーチン・プログラムとして形成したベクトルを、前記基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット：

(01)  $W04i = \Phi 4[\{L4\} + \{O4\} + \{S4\} + R4]i$

(02)  $W02i = \Phi 2[\{L2\} + \{I2\} + R2C + R2]i$

(03)  $W03i = \Phi 3[\{L3\} + R3R + R3C + R3D + R3M + R3E]i$

に集合させる第4のステップと、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数を用いて前記パレットをプログラム化する第5のステップとを具備することを最も主要な特徴とする。

[0010] 本発明の異なる実施体としての「開発対象のソフトウェア」を生産するためのプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、ソフトウェア開発管理装置は、開発対象のソフトウェアに関する要件を、要件をとらえる最小単位である主語＝[名詞, 狭義の属性, 広義の属性]での定義に係る名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、



これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉えるPRD手段(或いはPRD部)と、前記定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム):

L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E

を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで前記ベクトルをサブルーチン・プログラムとして形成するベクトル形成手段(或いはベクトル形成部)と、前記サブルーチン・プログラムとして形成したベクトルを、前記基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

に集合させるパレット手段(或いはパレット部)と、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数を用いて前記パレットをプログラム化するパレット連鎖関数手段(或いはパレット連鎖関数部)とを具備するように構成することもできる。

[0011] 本発明はさらに、上述の「開発対象のソフトウェアを生産する方法」によって生産されたソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現されるが、この場合の本発明は、予め規定された14種類のベクトル(サブルーチン・プログラム):

L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E

が、ベクトルの実行を管理するパレット関数を伴う3つのパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

に集合(パレット化)されて得られる全体に、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数が配置される構造を有するように構成することもできる。

。

[0012] ここで、「PRD(プロセス・ルート・ダイアグラム)手段」とは、名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉えるための機能を有する専用チップ・専用回路等の「PRD部」として、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現することが可能であり、本願発明はこれらの実施形態総てを網羅するものである。

[0013] 「ベクトル形成手段」とは、定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム):  
L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E  
を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで前記ベクトルをサブルーチン・プログラムとして形成するするための機能を有する専用チップ・専用回路等の「ベクトル形成部」として、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現することが可能であり、本願発明はこれらの実施形態総てを網羅するものである。

[0014] 「パレット手段」とは、サブルーチン・プログラムとして形成したベクトルを、該基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

に集合させるための機能を有する専用チップ・専用回路等の「パレット部」として、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現することが可能であり、本願発明はこれらの実施形態総てを網羅するものである。

[0015] 「パレット連鎖関数手段」とは、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数を用いてパレットをプログラム化するための機能を有する専用チップ・専用回路等の「パレット連鎖関数部」として、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現することが可能であり、本願発明はこれらの実施形態総てを網羅するものである。

[0016] またさらに本発明は、上述の「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現されるが、この場合のうちの14種類のベクトル(サブルーチン・プログラム)いずれかとしての本発明は、

L4(論理要素):未来の自己を表す名詞に真偽値を決定する為の機能を有する;

L2(論理要素):子孫を有しない自己を表す名詞に真偽値を定義する為の機能を有する;

L3(論理要素):過去の自己を表す名詞に真偽値を決定する為の機能を有する;

I2(入力作用要素):上記L2の成立を支援する為の機能を有する;

O4(出力作用要素):上記L4の結果を宣言する為の機能を有する;

S4(同期作用要素):ベクトルの領域を用いてSFの周期性を整備する為の機能を有する;

R4(経路作用要素):W04とW02に順序性を与える為の機能を有する;

R2C(経路作用要素):W02と隣下位のW04に順序性を与える為の機能を有する;

R2(経路作用要素):W02とW03に順序性を与える為の機能を有する;

R3R(経路作用要素):W03とW04に順序性を与える為の機能を有する;

R3C(経路作用要素):W03と隣下位のW04に順序性を与える為の機能を有する;

R3D(経路作用要素):W03と隣上位のW03に順序性を与える為の機能を有する;

R3M(経路作用要素):W03と隣々上位のW04に順序性を与える為の機能を有する;

R3E(経路作用要素):SFの作用を終了させる為の機能を有する;

のいずれかであることを特徴とする。

- [0017] また、上記「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現される場合のうちのベクトルの実行管理を行うパレット関数(サブルーチン・プログラム)としての本発明は、

$$(01)W04i = \Phi 4[\{\{L4\} + \{04\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

のいずれかの集合化と其の網羅とを行う $\Phi 4$ 、 $\Phi 2$ 、 $\Phi 3$ のいずれかの機能を構造として具備することを特徴とする。

- [0018] 上記「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現される場合のうちのパレット関数の実行管理を行うパレット連鎖関数(プログラム)としての本発明は、3種のパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{04\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

で構成されるSFに対して、

$$SF_i = \Phi 0[W04 + W02 + W03]i$$

で規定される $\Phi 0$ の構造を具備することを特徴とする。

- [0019] さらに、本発明は、上述の「開発対象のソフトウェアを生産する方法」による、要件から抽出した情報(ドキュメント(紙、データ))たる名詞情報、入出力定義体情報、PRD情報のいずれかの情報の抽出方法として、またかかる抽出方法によって抽出された情報(ドキュメント(紙、データ))として、さらには当該抽出された情報の使用方法として、或いは、これらの情報が搭載された情報記録媒体として、または情報の抽出方法／使用方法がコード化されたソフトウェア、当該ソフトウェアが搭載された記録媒体／

装置(ハードウェア)として、いずれも実現することができるが、この場合の本発明は、要件から抽出すべき情報として、

名詞情報:名詞=[名称, 識別子, 所定の22種のいずれかに分類される名詞種別]により定義されるべき情報;

入出力定義体情報:入力定義体、出力定義体、並びにそれらに属する名詞の関係に関する情報;

PRD情報:所定の規則によってSFを順序づける情報を備えることを特徴とする。

[0020] 本発明における思考モデルは電算機上で定義される。

[0021] 換言すれば、この思考モデルは電算機上で定義可能となるように定義される。

結果から言えば、この思考モデルを定義する仕方は電算機のソフトを開発する場合の思考法を反映する。其して、本論で述べている様に、其の定義の仕方には公式と見なせる普遍性が成立する。

[0022] これは本研究の成果である。

[0023] この普遍性の証明は、結果的に添付されているベクトルの定義規則(資料1), T M P(資料2)で与えられる。

[0024] 其処には例外のない規則が論述されているだけである。

本研究では、思考は自己に成立する意識の作用として定義される。其して、この作用をモデル化する為に本研究では存在論をモデル化する。其れは存在, 意図, 自己, 意識をモデル化する事である。

本研究では、思考は存在論の部分を成す作用であると考え。其の為、存在論の全体をモデル化する必要がある。そうでなければ、論理的に妥当な思考モデルを定義する事が出来ないからである。本研究が長期にわたる時間を要した理由はこの為である。

本研究の存在に関する思想性は基本概念(第2章2. 3), 存在の概念(第2章2. 4. 1)で明示されている。其して、本研究では其れらを裏付ける為に集合論の概念を用いて、モデル化が行われる。

本研究では、存在は連鎖として定義される。其して、連鎖の要素は論理原子である



。論理原子は概念であり、その概念は本研究で仮説されている。其して、論理原子は本研究で仮説される時間速度から導かれる。

本研究では、存在は集合の概念を用いてモデル化される。この為に、時間速度、並びに論理原子は重要な位置を占める。

[0025] 時間速度は、限定的で無次元の有理数である。其して、本研究では存在が空間を占める概念であると考え事から、論理原子は空間を表す有理数として位置付けられる。

本研究では、意識は自己に内在する連鎖である。

[0026] 其して、本研究は、思考を以下の様に定義する。即ち、意識に内在する連鎖が自己を介して開放される様相が思考の作用である。

本研究では、思考の作用を脈絡と定義する。脈絡に属す全ての連鎖が言表出来れば、其の全言表の集合は其の思考の限界を越える作用を成立させる事になる。これが、本研究で行われる存在論に関する論考の結論である。

本研究では、叙述と言表の用法を区別している。叙述は存在の定義を意味し、言表は存在が存在する事の証明を意味する。叙述は時間的に過去の存在を定義する事が出来る。

[0027] しかし、言表は時間的に未来の存在しか証明する事が出来ない。

脈絡に属す連鎖の成立を証明する為には、其の連鎖の過去と未来の連鎖の言表が成立する事が必要である。何とならば、自分で自分を証明する事は出来ないからである。其れ故、今の連鎖の過去の連鎖が未来の連鎖と同じ位置付けになる様に、脈絡の構造は修正される。其れがENWである。其して、言表の構造がPSである。ENWの全体を言表する構造がTDMである。思考モデルの仮説を電算機を用いて実証する基本構造がSFである。其して、其れを一般的に実証する構造がPRDである。ENWの全体を言表する構造は、TDMから与えられる二つの制御プログラム( $\Phi 0$ ,  $\Phi P$ )である。これらプログラムには、普遍性が成立する。

PSはSF, PRDにおいては要件を捉える最小単位に変わる。其れがベクトルである。ベクトルの定義の仕方は普遍的な規則になる。其れをプログラム言語で言表したものがTMPである。

[0028] 即ち、電算機上で実証される思考モデルは二つの普遍的な制御プログラム、ベクトルで定義する事が出来る。其して、其の思考モデルは要件を満たす役割を電算機上で果たす。

[0029] 其の働きが妥当であるならば、本研究における仮説もまた、妥当性であると考えられる。

本方法で定義されるプログラム(SF, またはPRD)は、SFの定義から其れを実装する工学的環境、即ち、入出力作用要素を除き、定義される。其れは、従来の方法によりプログラムが成立する位置と異なるものである。其れを図21に示す。

[0030] これは、本方法論によるプログラムの役割が従来のプログラムの役割と必ずしも同じでない事を意味する。本方法論によるプログラムは、要件を捉えるプログラムとして位置付けられる。従来のプログラムは、要件をプログラムで再定義する役割として位置付けられる。即ち、従来のプログラムは要件に矛盾がない事を前提に定義される事になる。其の為に、事前に要件の妥当性を検証しておく必要がある。

[0031] 其れに対し、本方法論によるプログラムは、従来のプログラムと同じ様に電算機上で可動可能であるが、要件の矛盾を発見する仕組になっている。付言すれば、本方法論のプログラムのソースライン数は従来のプログラムのソースライン数に較べ概算7倍〜10倍となるが、其の役割の中には従来のプログラムが担っていない要件の矛盾を発見する仕組も含まれているのである。

本研究では、意味とは同期状態において新たな存在が定義される作用の事だと定義する。

其して、其の作用は意識の世界で成立すると考える。

本研究では、この作用をモデル化する。付言すれば、このモデルが本研究で言うソフトを定義する為の役割を果たす。

### 発明の効果

[0032] 本研究の成果は、この思想に基づいて論考された結果である。例えば、要件の曖昧さを回避して要件を定義するベクトルは、われわれの思考法を成立させる最小単位の発見に匹敵する概念である。この事によって、以下の問題を克服する原動力に成ると思われる。

(01)われわれの思考法に生じる曖昧さの問題

(02)論理の中に潜む恣意的な問題

(03)言表不可能な事態に対処する問題

其れは、本研究の成果を用いて既に開発されているシステムを考察すれば、其の事を予感する事が出来る。正常な評価は、現在の状況では成し難いと思われるが、本研究の成果である概念を認識すれば、其の効果も容易に認識する事が出来る筈である。

本研究の成果であるPRDを考察すれば、われわれの論理的史観の深層に位置する性質を捉えている事が分かる。この性質は、論理的史観を前提とする立場で捉える事は困難であるかもしれない。しかし、既存のプログラムから普遍的なアルゴリズムでPRDを捉える事が出来る。この場合のこのアルゴリズムは、本研究の成果から得られるものである。即ち、本研究の成果は、既存のソフトの世界とは無縁ではなく、深層において繋がっている。この事は、本研究の成果が既存のソフトの世界の課題を克服する為に用いる事が出来る事を示している。

以上の本研究の成果は、ベクトルの定義規則が種別化された主語(要件に属す名詞)により決まり、其の定義規則に従えば、最小クラスの機能が求められる。其して、主語の種別数、ベクトルの定義規則の数、其れによって決まる最小クラスの機能の種別は、全て既定値となる事によるものである。

付言すれば、要件を捉える為のわれわれの思考法に改善を齎す。例えば、要件に関する関係者の認識から曖昧さを除く事が出来るので、能率的に要件に関する合意を作り出すことが出来る。ソフトの世界に最小クラス概念を導入する事は、要件を捉える為のわれわれの思考の単位となり得るだけでなく、プログラムを定義する1命令文までを公共化する効果を獲得する事が出来る。

発明を実施するための最良の形態

[0033] 題目:ソフトウェア構築の摂理に関する統一理論の研究

Research on Methodological Universal Framework for Software Development

要旨

本研究の狙点は、ソフトをプログラム言語で述定する為に開発要件をアルゴリズム化する公式を求める事である。

例えば、われわれが言葉を発する直前迄、発する言葉を記憶する事が出来ないにも拘らず、われわれは言葉が無意識的に紡いで文章にしている。

本研究では、これと同じ原理でプログラムが述定出来ないだろうかとして論考される。

この為に、本研究では「存在」「意図」「意識」に関する論考を形而上学的に行う。其して、其の結果、ソフトは「意識を成立させる外延的な存在を内包する作用」として定義される。

[0034] 其して、この定義から誘導される新たな定義により、プログラムを定義する公式が求められる。

本論文はこの研究に関する報告である。

Keywords:

仮想存在, 仮想空間 (TDM), 自己なる存在の構造, 意識を表す存在, 存在の証明, 言表の構造, 仮想空間の実装構造, 要件を捉える公式。

本研究を始める動機は、ソフトに対するわれわれの考え方の不自然さに対する疑問である。其れ故、本研究の前半は其の動機の精神性を現象的に検証する事に費やされ、プログラムの分析、開発への参画、開発工程の分析、論文調査、プロダクトの分析等が行われた。後半は、其の不自然さの原因を克服する思考方法を求める事であった。

目次

[本発明の思想的背景]

第1章 序論

1. 1 思考の曖昧

1. 2 思考の限界

1. 3 本研究の狙点

第2章 仮想存在

2. 1 仮想存在の連糸

- 2. 2 言表
- 2. 3 基本概念
- 2. 4 存在の構造
  - 2. 4. 1 存在の概念
  - 2. 4. 2 連鎖の概念
- 2. 5 不可知空間
- 2. 6 IDS
- 2. 7 連鎖
  - 2. 7. 1 意識連鎖
  - 2. 7. 2 確立連鎖
  - 2. 7. 3 事象連鎖
  - 2. 7. 4 多重連鎖
  - 2. 7. 5 群化多重連鎖
  - 2. 7. 6 自然連鎖
  - 2. 7. 7 群化自然連鎖
  - 2. 7. 8 仮想存在の属性に関する仮想の為の思想
- 2. 8 自己なる存在のモデル
  - 2. 8. 1 臨界連鎖
  - 2. 8. 2 特異連鎖
  - 2. 8. 3 意図のモデル
  - 2. 8. 4 意識のモデル
- 2. 9 思考モデル
  - 2. 9. 1 脈絡の定義
  - 2. 9. 2 客体化
  - 2. 9. 3 陳述
  - 2. 9. 4 同化
  - 2. 9. 5 記憶 34

### 第3章 仮想空間(TDM)



- 3. 1 存在の位置
- 3. 2 言表の仕組
- 3. 3 言表の構造
- 3. 4 脈絡構造の再定義
- 3. 5 存在の証明
- 3. 6 TDMの定義

[本発明の内容]

#### 第4章 仮想空間の実装構造(SF)

- 4. 1 要件
- 4. 2 主語
- 4. 3 ベクトルの原型
- 4. 4 ベクトルの集合
- 4. 5 ベクトルの定義規則
- 4. 6 パレット関数
- 4. 7 パレット連鎖関数
- 4. 8 パレット
- 4. 9 SF

#### 第5章 要件を捉える公式(PRD)

- 5. 1 PRDの構造
- 5. 2 SFの結合規則
- 5. 3 端点と始点の関係
- 5. 4 入力定義体と出力定義体の相補関係
- 5. 5 SFの縮約規則
- 5. 6 ベクトルの座標
- 5. 7 テンプレート(TMP)の意義
- 5. 8 適応化の幾つかの工夫 5
- 5. 9 本方法論の総括

## 第6章 結論

本論文で用いる図・表・参考資料一覧

図1 :IDS

図2 :論理原子の空間速度

図3 :占有速度

図4 :空の連鎖

図5 :単元の連鎖

図6 :連糸

図7 :臨界連鎖

図8 :特異連鎖

図9 :意図

図10:意識

図11:自己なる存在のモデル

図12:脈絡のモデル

図13:存在の証明

図14:未来の自己の言表構造(PS4)

図15:過去でも未来でもない自己の言表構造(PS2)

図16:過去の自己の言表構造(PS3)

図17:未来の自己が存在しない自己の言表構造(PS2-1)

図18:ENWのモデル

図19:TDMのモデル

図20:要件の捉え方

図21:プログラムの位置付け

図22:ベクトルの標準形

図23:SF(基本構造)

図24:PRD

図25:パレット連鎖関数の標準形

図26:パレット連鎖関数の標準形

図27:パレット関数の標準形

表1 :名詞種別と主語種別(ベクトル種別)の関係

表2 :名詞の種別

表3 :名詞の狭義の属性

表4 :ベクトルの原型

表5 :S4ベクトルの主語の種別

表6 :ベクトルの原型と領域の関係

表7 :名詞の広義の属性

表8 :経路制御情報テーブル

表9 :開発実績一覧表

資料1:ベクトルの定義規則

資料2:テンプレート(プログラム言語がVBの例)

[本発明の思想的背景]

## 第1章 序論

ここでは、本研究の目的を得る為に強い影響を受ける2つの問題、思考の曖昧さと思考の限界の問題が整理される。其して、本研究の目的が明らかにされる。

### 1.1 思考の曖昧

われわれの言表行為は、言表不可能な意識を反映している。

[0035] しかし、言表行為にも限界がある。其の意味で、われわれの言表行為は意識を十分に反映しているとは考え難い。

ヴィトゲンシュタインは、言葉の意味を其の言葉の使われ方として捉えている(参考文献14参照)。本研究では、言葉の意味をヴィトゲンシュタインの立場では論じない。

[0036] 其の代わり、われわれが用い得る言葉の数の概念を論じる。其して、其の言葉の数は時間と共に増加すると考える。

[0037] 本研究では、この限界の問題を次の様に解釈する。即ち、われわれが今使える言葉の数を $n$ 個とすれば、われわれは其の数を越えて言表を行う事が出来ない。

他方、われわれの言表行為は時間の流れの中で逐次的に行われるので、其の行為は外延的に成立する。即ち、われわれの言表行為は $n$ 個の言葉を用いて未来の事態を定義する関係になっている。

しかし、本研究では、未来では言葉の数は経時的に増加し、われわれが言表した時点では其の言葉の数は既に $n$ 個よりも大きくなっていると考ええる。

[0038] 例えば、其の時点での言葉の数を $m$ 個とすれば、われわれは、本来、 $m$ 個の言葉で定義されるべき事態を $n$ 個の言葉で言表している事になる。

[0039] この言葉の数の違いが、言表に内在する曖昧さとしてわれわれに啓示されると考えられる。何となく釈然としない事態が自己に引き起こされている事である。

[0040] しかし、このような性質の問題をわれわれの言表で克服する事は不可能である。

われわれの言表行為は、上述の様に外延的に成立している。換言すれば、未来に向かって成立している。其れ故に、過去の事態を言表する事は不可能である。

[0041] 譬え、今朝の出来事でさえ、其れを言表する瞬間に其の出来事は未来の事態として述定されてしまう事になる。

また、われわれは、経験から未来を予測する事が不得手な事はわかっている。しかし、われわれの言表は未来を予測する事に似ている。しかし、どの程度の未来の述定であるかは不明である。

[0042] 其れは、言表出来ないからである。

[0043] 其処で、本研究では、言表はほんの一瞬の未来を述定するのだと考える。

[0044] 其して、われわれの言表では未来とは言えないほどのほんの一瞬の未来が述定されているのだと考える。其の限りにおいても、われわれの言表には曖昧さが生じていると推論する。

例えば、通常、厳密性があると思われるプログラムのロジックにも曖昧さが内在する事を拙著の論文[Ambiguity hidden in traditional programs and meaning of Vectors of LYEE "(pp.200-213) NEW TRENDS IN SOFTWARE METHODOLOGIES, TOOL AND TECHNIQUES (Volume98), " Edited by Hamid Fujita and Paul Johannesson

,IOS Press]で明らかにしている。”

本項の主張を裏付けるものとして提示する。

## 1.2 思考の限界

本研究では、われわれが行う言表は思考作用の結果として位置付けられる。其して、其れらは新たな存在を定義する為に行われる作用であると推論する。其の作用の源は意識であると推論する。

[0045] 即ち、言表は意識を反映する作用として成立する。この前提に立てば、われわれが行う言表は意識を十分に反映しているかどうか、という疑問がある。

[0046] われわれの言表がどの程度、意識を反映しているかは自己も含めて誰も分からないからである。

本章1.1で述べる様に、言表の為に其の時点で用いられる言葉の数 $m$ 個を認識する事が出来ないからである。要約すれば、われわれは恒常的にいつも言葉不足の中に置かれている。

[0047] 換言すれば、われわれの思考は新たな事態を創造する為に行われるのではなく、既に成立している存在から既に成立している存在を述定する為に行われている様に思われる。

[0048] 創造される存在は、われわれの思考とは別のところで定義されているに違いない。これが、本研究の基底に置かれる思想である。

## 1.3 本研究の狙点

われわれが言表不能な事態に遭遇すれば、其れがわれわれの思考の限界である。

[0049] 例えば、言葉の発報者は発報前の事態、並びに発報する言葉を認識する事は出来る。しかし、其の言葉を発報する直前の事態は認識出来ない。この事は、認識可能な事態と発報直前の事態が異なる成立の仕方をするからだと考えられる。

本研究の目的は、発報直前の認識不可能な事態をモデル化する事にある。其れは、本章1.1の言葉の数の変化の仕組をモデル化する事と同じになると想定される。



[0050] 其して、其の仕組のモデルは、意識を表す存在と其れを反映する存在の関係を論じなければならない事は明らかである。其の事が、本章2. 2の思考に内在する問題を克服する手立てを其の仕組のモデルに取り入れる論拠を与えるであろうと推測されるからである。

この為に、本研究では存在を仮想(図1)し、其れを基に上述の仕組のモデルを定義する為に以下の仕組を論考する。即ち、

(01)思考を成立させる自己なる存在の仕組

(02)上記(01)の仕組を基に思考の限界を克服する仕組

(03)上記(01)(02)の意義を実証する為の仕組

結果的に、上記(01)は図11に、上記(02)は図19に、上記(03)は図23, 24に帰結する。それぞれの図の説明は、それぞれの章で行われる。

ソフトは、言葉を基に電算機を用いて新たな言葉を利用者に成立させる事態として定義する事が出来る。(因に、言葉の意味は人に宿り電算機には宿らない)

本ソフトの定義は、上記(02)の思考の限界を克服する仕組(図19)でソフトが定義出来る事を意味する。

[0051] この場合、この仕組で定義されるソフトは、要件が成立する限界の事態を要件として成立するものである。其れ故、本ソフトはこれ迄のソフトとは認識上の位置が異なる。

[0052] 因に、図21参照。

[0053] 本ソフトを従来のソフトと同じ役割を果たさせる為には、認識上の位置を従来のソフトの位置に移動させる必要がある。本ソフトは、其の作用を電算機で行わせる。

この事から、上記(02)の仕組は本ソフトの定義が容易に行える様に、また、電算機の役割が能率的に果たせるようにする為に、上記(03)のモデル(図23, 24)に書き換えられる。

結果的に、上記(03)の仕組は要件を定義する公式に帰着する。この公式の意味は、数学で言う公式の概念と同義である。

本ソフトと従来のソフトを定義作業の能率性の観点から比較すれば、其れは、上記(01)(02)の論考の意義、並びに仮想の妥当性が検証出来る事になる。即ち、本ソフトの定義作業が従来のソフトの定義作業に較べ、能率性の改善性が認められれば、其

れは本研究の仮想に妥当性があると考えられるからである。

上記(03)の仕組を用いて、これ迄に開発され、稼動しているソフトを表9で示す。

- [0054] この中には、開発終了の後、周辺関係者の思惑や風聞に晒され、従来のソフトに置き換えられたと報告されているものが1件ある。

これら開発事例を用いて、本研究の意義を科学的に評価する事は可能である。

- [0055] しかし、其れはまだ行われていないので、今後の課題である。

## 第2章 仮想存在

本研究では、自己なる存在が仮想される。其の為に、様々な仮説が行われる。

- [0056] 本章では、其の仮説を述べる。

### 2. 1 仮想存在の連糸

自己なる存在は過程を得て成立する。其の過程を、本研究では存在の連糸(以下、連糸)と記す。連糸は、段階的に定義される存在により段階的に作り上げられる。

本研究では、どの段階に於いても、其の段階を成立させる為の存在を定義する作用が確率的に中断させられる事である。これは、本研究で仮説されるもののひとつであるが、其の中でも基本を成す概念である。其れは、われわれの世界における全ての存在が時間的に移り行く様相を見せる事を、本研究のモデルの上にどのように表すかと言う論点から設定されている。連糸を作る段階で、中断させられる事態が生じれば、其れまでの連糸は破棄され、最初の段階から改めて作り直される。

連糸は、其れを成立させる為の存在の定義の仕方により、経緯、群化、脈絡の段階に分けられる。換言すれば、存在は、経緯、群化、脈絡により定義の仕方が異なる。

経緯では、自己なる存在を導く元になる存在が定義される。

- [0057] 群化では、其の元になる存在から自己なる存在に至る存在が定義される。この過程で、自己なる存在が定義される。

- [0058] 其して、脈絡では、自己なる存在に統治される存在が新たに自己なる存在として定義される。本研究では、この事を意識を表す存在が新たな存在を解き放すという言い方をする。

経緯の段階では、連糸の1本を紡ぐ存在が成立する。其の存在を本研究では確立

連鎖, 事象連鎖と記す。其して, この段階の最終局面では, 其の1本の連糸を徹底的に解す為の存在が定義される。この存在を本研究では多重連鎖と記す。連鎖の詳細は, 本章2. 7で述べる。同時に, 其の最終段階では其れらを統治する作用が生じる。この作用を, 本研究では群化と記す。群化の詳細は, 本章2. 7の2. 7. 5と2. 7. 7に述べる。

群化により, 解される存在(多重連鎖)を統治する存在が成立し, 中には其れら存在を限界まで統治する存在が成立する。この存在を, 本研究では特異連鎖と記す。特異連鎖は, 自己なる存在を成立させる候補である。

自己なる存在を図11でモデルとして表す。詳細は, 本章2. 8で述べる。

記述の様に, 脈絡として自己なる存在が定義されれば, 其の存在は直ちに自己を介して統治している存在を新たな自己なる存在として解き放つ。

[0059] 脈絡の過程で中断が生じなければ, 脈絡の停止条件に遭遇する迄, 其の作用は継続される。

[0060] 脈絡の停止条件に遭遇すれば, 連糸を定義する最初の段階である経緯の初めに戻る。

[0061] 図6は, 連糸の構造を示す。本図では自己なる存在は省略されている。また, 図中の意識 $\lambda$ 集合の列, 認識 $\lambda$ 集合の列, 開示, 連想の概念は, 本章2. 7で述べる。

## 2. 2 言表

本研究では, 存在は主語と属性で定義される。これを叙述と記す。この叙述は, 其の存在が成立している事を証明するものではない。本研究では, 存在が存在している事の証明は, 言表によるものとする。其の言表の構造を, PSと総称する。即ち, PSの叙述が成立すれば, 其の事が言表となり, 其の言表により其の存在の成立が証明される。

[0062] PSについては, 第3章3. 2と3. 3で述べる。其れに関係する図は, 図14, 15, 16, 17参照。

存在Aと其の叙述aをA(a), 存在Aを統治する存在Bと其の叙述bをB(b)とする。この場合, 存在A(a)は解き放される存在, 存在B(b)は存在A(a)を統治する意識を表

わす存在とする。解放される存在の詳細は、本章2. 9の2. 9. 1で述べる。

其して、存在Bを統治する存在Cと其の叙述cをC(c)とする。この存在C(c)は、本研究では自己なる存在を代表する存在である。この存在の詳細は、本章2. 8で述べる。

[0063] この場合、本研究のモデルではAがBとCを介して解き放される。解き放される構造を、言表と記す。其して、存在A, B, Cの間には、以下の関係が成立する。

[0064] 即ち、 $C(c) \supset B(b) \supset A(a)$ である。

解き放される存在には、上記の関係 $C(c) \supset B(b) \supset A(a)$ の基で以下の(01)～(04)の関係が成立する。即ち、

(01) $C'(c') \supset B'(b') \supset A'(c' \supset b' \supset a')$

(02) $C(c) \rightarrow C'(c')$

(03) $B(b) \rightarrow B'(b')$

(04) $A(a) \rightarrow A'(c' \supset b' \supset a')$

本来的に存在の連糸を構成する存在の叙述は外延的である。其れは、叙述a, b, cの間に大小の関係がないと言う意味である。

[0065] しかし、解き放される存在Aの叙述a'には、上記(04)の関係により、大小関係が成立してしまう。

[0066] 即ち、a'がb', c'に内包化されてしまう事である。この関係の元で叙述されるa'が、本研究で言うA'の言表である。

[0067] 本研究では、言表が成立する存在を物質的存在と記す。其して、叙述に留まる存在を非物質的存在と記す。

## 2. 3 基本概念

仮想存在(以下、存在)を定義する為の基本概念を以下に示す。

(01)存在は、固有の空間である。

(02)其の空間には、其の空間の定義と其の空間が成立する仕組が叙述されている。

(03)空間の定義は、其の存在の性質を示す。

(04)空間が成立する仕組の定義は、其の存在の役割を示す。

- (05)同じ役割の存在は、同じクラスを作る。
- (06)存在が成立する仕組には、終局がある。
- (07)空間の定義の叙述が失われる事がある。
- (08)空間が成立する仕組の定義の叙述が失われる事がある。
- (09)存在の叙述が失われる事がある。

本研究では、存在は主語と補語で定義される文章に例える事が出来る。其して、既に成立している文章で、其れらのどちらかが失われれば、其の文章は成立しなくなる。本研究では、これを存在の叙述が失われると言う。

[0068] 本研究では、存在は連鎖として定義される。其れ故、この事態は既に成立している連鎖において生じる事になる。連鎖については、本章2. 7で述べられる。

## 2. 4 存在の構造

本研究では原理的な空間が仮想される。其の空間を空間速度と記す。其れを成立させる要因を論理原子と記す。論理原子は、本章2. 6で述べる。仮想される存在はこの論理原子を用いて定義される。仮想される存在を連鎖と記す。連鎖は、本章2. 7で述べる。本節では、通常の存在の概念と連鎖の対応関係を示す。

### 2. 4. 1 存在の概念

スピノザの思想は、例えば、人が想像力によって理解出来ない事態は形而上学的な仕方でのみ理解されるべきであると言うものである(非特許文献15参照)。其の為にスピノザは、存在を論考するに際し形而上学的な仮説を公理として定め、存在の理解を果たそうとしている。本研究は、思考の限界をモデル化する立場から、スピノザが論考しようとする立脚点と同じ立場に立つとしている。

上述の1. 3の基本概念は、存在についての仮説であり、其れをモデル化する為の概念、即ち、本研究では公理となる概念がここで示される。

- (01)存在は、属性と其れを基に決まる実体とで構成される。
- (02)属性は、存在の役割を示す。
- (03)存在の役割は、既に約束されている。
- (04)実体は、存在を代表する。



(05)実体が成立しなければ、其の属性は無役となる。

(06)存在は、属性が実体を介して属性の性質を叙述する。

(07)実体と属性が成立するクラスは異なる。

(08)属性を決定する規則により実体のクラスが定義される。

#### 2. 4. 2 連鎖の概念

本研究では、仮想される存在を連鎖と記す。連鎖は、本章2. 4. 1の存在の概念を基に定義される。即ち、

(01)属性は論理原子の集合である。

(02)実体は論理原子の集合を属性に変える1個の論理原子の事である。論理原子が表す空間を空間速度、論理原子の集合が表す空間速度を占有速度と記す。占有速度の概念を図3で示す。

(03)実体は、論理原子の集合の占有速度に近似する空間速度を有す論理原子が選ばれる。

(04)論理原子の集合が実体を選ぶ其の選び方には、占有速度の外側から選ぶ道筋と内側から選ぶ道筋がある。

01. 前者の場合、実体は其の集合に属す以外の論理原子から選ばれる。この場合、其の属性は自己が部分である事を実体を介して叙述する。この属性と実体の関係を単元と記す。

02. 後者の場合、実体は其の集合に属す論理原子の中から選ばれる。この場合、其の属性は自分が全体である事を実体を介して叙述する。この属性と実体の関係を空と記す。

#### 2. 5 不可知空間

本研究では、根源的な要素と其の集合が仮説される。其の要素を時間速度( $V_i$ )、其の集合を不可知空間と記す。不可知空間に同じ時間速度は属さないものとする。

本節では、不可知空間に属す時間速度を元に成立する概念について述べる。不可知空間と時間速度は、論理原子を成立させる為の前段の役割を果たす。

(01)時間速度の値は其の値の活力となる。

01. 時間速度の値は+値である。

(02)不可知空間の部分集合は成立しない。

02. 不可知空間の活力は不可知空間に属す時間速度の総和である。即ち,  $\sum 1 \phi Vi$  で表す。

03.  $\sum 1 \phi Vi$ は定数である。

04.  $\sum 1 \phi Vi$ は分割される事はない。

05.  $\Phi$ は不可知空間に属す時間速度の総数で定数である。

06.  $\sum 1 \phi Vi$ は時間速度の最大値よりも大きい。

07.  $\sum 1 \phi Vi$ は時間速度ではない。

08.  $\sum 1 \phi Vi$ の逆数( $2 \epsilon$ )は定数である。

09.  $2 \epsilon$ の逆数は不可知空間の空間の大きさを表す。

10.  $2 \epsilon$ は最小の時間速度を表す。

11.  $2 \epsilon$ は不可知空間に属す時間速度の順序列の共通差分を表す。

12. 値  $\epsilon$ は定数でゼロを代理する。

13. 値  $\epsilon$ の逆数( $1/\epsilon$ )は定数で最大の自然数である。

14.  $\Phi$ と $1/\epsilon$ は一致する。この自然数を臨界自然数と記す。

15.  $\sum 1 \phi Vi$ は臨界自然数よりも小さい。

16.  $2 \epsilon$ の倍数の逆数は時間速度になる。

(03)時間速度とは上記01ー16を条件として満たす値の事である。

(04)時間速度は果敢なく逐次無作為に選ばれ, 複写されて対となる。

[0069] この作用の為の活力は, 不可知空間の活力で賄われる。

(05)複写される時間速度の片方は直ちに不可知空間の補集合領域に移る。

[0070] この作用は, 移る時間速度の活力を用いて行われる。

本研究は, 序論で述べた様に, 思考の曖昧さや思考の限界の影響を受ける課題である。換言すれば, 本研究は, 無限性の問題を議論せざるを得ない立場に立たせている。其の為に, 不可知空間に属す限定的な有理数となる時間速度の値  $\sum 1 \phi Vi$ は, カントールの連続体仮説, 或いは, ラッセルの逆説に触れる事になる(非特許文献13参照)。この問題を回避する為に, 本研究では, 上記(02)05, 12が定義される。この定義の妥当性については, 本稿ではこれ以上触れない。

## 2. 6 IDS

IDSとは、不可知空間の補集合領域に属す時間速度を要素とする集合である。図1にこの概念を示す。

本研究では、自己なる存在が成立すれば、其処に属す意識を表す存在が作り出す存在の脈絡を思考として定義する。其して、本研究で言う言表は、脈絡を成す存在により成立する。しかし、其の源となる意識を表す存在に、言表を成立させる事は出来ない。

[0071] 言表を成立させる事が出来ない意識を表す存在が、本研究で言う思考の限界を示す存在である。

本研究の目的は、この意識を表す存在を言表する仕組のモデルを定義する事に帰着する。この目的の為に、存在が仮想される。其して、其の仮想される存在を連鎖と記す。其の連鎖を定義する為には、更に其の前段を成す仮説が必要になる。IDSは、其の仮説を述定する為のモデルの事である。

[0072] 図1に記される用語は、定義される仮説上の概念の称呼である。其の詳細は適宜本論文中で述べられる。

IDSは、不可知空間に倣い同じ値の時間速度を其の要素とする事が出来ない。即ち、不可知空間から到来する時間速度が重複する事になれば、其の場合、別のIDSが作られる。其れ故、IDSは唯ひとつとは限らない。本研究では1個のIDSが論考される。

IDSの論考から連鎖を成立させる前段の概念が定義される。論理原子も其の概念のひとつである。本節では、IDSに成立する概念を以下に示す。

### (01)境界時間速度:VB

IDSに属す時間速度の個数は、不可知空間で行われる時間速度の複写により、逐次、増加する。其の個数が3個を越えれば、其の都度、時間速度の値の順序列が定義し直される。其の順序列の時間速度を大なる方から数えて、中点に位置する時間速度は其の時点でIDSに属す時間速度を代表する。この時間速度を境界時間速度と記す。

(02)時間速度( $V_i$ )の位置活力:  $|V_i - V_B|$

IDSに属す時間速度は、この概念で相対化される。

(03)境界原子の位置活力:  $\epsilon$

本研究では、全てが存在として位置付けられる。例えば、数学的ゼロも存在と認められない。其の為に、数学的ゼロは  $\epsilon$  に置き換えられる。

(04)論理原子の空間速度:  $1 / |V_i - V_B|$

空間速度とは体積的空間の広がりである。其の尺度が空間速度と呼ぶ値である。空間速度と同じ値の時間速度がIDSに存在する場合に限り、空間速度は成立する。

(05)境界時間速度の空間速度:  $1 / \epsilon$

この値は定数で、且つ最大の値になる。其れ故、この値の空間速度は、最大の空間速度になる。この事から、境界原子の空間速度はIDS其れ自体を代表する空間速度となる。境界時間速度が入れ替わっても、この空間速度は定数となる。

(06)論理原子

時間速度、位置活力、空間速度の3種の値で定義される概念である。境界時間速度を含め、其れよりも大きい時間速度を有する論理原子を意識原子、境界時間速度より小さい時間速度を有する論理原子を認識原子と記す。

[0073] 特に、境界時間速度の論理原子を区別する必要があるれば、境界原子と記す。

[0074] 図2に論理原子の空間速度を示す。論理原子とは、連鎖を叙述する為に用いられる要素である。

(07)存在線

不可知空間と論理原子が着座するIDSで成立する座標の集合が存在線である。論理原子になり得ない時間速度は、存在線に着座する事が出来ない。其れら時間速度は単にIDSに浮遊する。

新たな時間速度が不可知空間から到来すれば、境界原子が定義し直される。其の為に、存在線に着座している論理原子は、時間速度に戻りIDSに浮遊する。

[0075] 其れは存在線の消滅を意味し、同時に着座していた不可知空間は、其のIDSから去る。其の後、境界原子、並びに論理原子の再定義が行われる。其して、存在線が

成立すれば、不可知空間は再来して着座する。

[0076] 新たな時間速度が不可知空間から到来する都度、存在線の消滅と再定義が繰り返行われる。

(08)不可知空間の座標

存在線が成立すると、不可知空間は境界原子の時間速度の値を座標として存在線に着座する。

(09)境界原子の座標

不可知空間は、境界原子の時間速度を座標として存在線に着座する。境界原子は、自己の時間速度の値  $\varepsilon$  が加えられた値を座標として存在線に着座する。

(10)論理原子の座標

意識原子の存在線における座標は、境界原子の座標を共用する。しかし、意識原子の空間速度の値は変わらない。認識原子は、それぞれの時間速度を存在線の座標として着座する。

(11)回帰と転位

境界原子の再定義が行われる事により、其れ迄の境界原子が認識原子に、或いは其れ迄の認識原子が境界原子に入れ替わる事がある。前者を転位、後者を回帰と記す。

[0077] この概念が連鎖(存在)が解体される条件になる。詳細は連鎖の所で述べる。

(12)論理原子の複写回数

存在線に属す論理原子は、連鎖の実体となる場合に複写される。

[0078] 不可知空間で複写される片方の時間速度が、其処に止まる事が出来ない為に其の補集合領域(IDS)に移されるように、存在線に属す論理原子が連鎖の実体となる場合、其の論理原子は複写される。其の複写される論理原子は、IDSに止まる事が出来ない。其の為、其の論理原子は其の活力を用いてIDSの補集合領域に移る。

[0079] この場合の複写回数には上限があり、其れを超えて複写される事はない。

[0080] 因に、不可知空間に属す時間速度の複写回数には制限がない。

本研究では、複写回数の上限の値を次の様に想定する。即ち、IDSは、新たな論理原子が到来すると、其の都度、其の新たな論理原子を含め、全ての論理原子に対



し複写回数の上限の値を付与する。其の値は、IDSに属す全ての論理原子の到来順位を逆に振り替えたものである。

認識原子で定義される連鎖には、6種族(確立連鎖、事象連鎖、多重連鎖、群化多重連鎖、自然連鎖、群化自然連鎖)がある。其して、同じ種族の連鎖の中には、実体が重複する物がある。この理由により、認識原子には複写回数の上限を越える事態が生じる場合がある。他方、意識原子で定義される連鎖には実体が重複する事がないので、意識原子には複写回数の上限を超える事態は生じない。連鎖に関する詳細は、本章2.7で述べる。

## 2.7 連鎖

連鎖は、属性が実体を介して行う属性の叙述の事である。

[0081] 其れは、論理原子を用いて定義される仮想存在の事である。連鎖の属性とは、論理原子の集合である。この集合は存在線上で定義される。

実体として選ばれる論理原子は複写される。其して、IDSの補集合領域に移る。この補集合領域を、(01)意識空間、(02)確立空間、(03)事象空間、(04)多重空間、(05)自然空間と記す。

[0082] 本章2.6(012)で触れた6種族の連鎖の実体は、この空間の要素になっている。

これら空間は、境界原子の空間速度の外側(補集合領域)に成立する。其して、意識空間は瞬きの様に成立と消滅を繰り返しながら次第に膨張する。意識空間については、本章2.7.1で述べる。

[0083] 存在の属性は存在線上で成立し、実体は上記の空間で成立する。即ち、実体と属性は、異なる2つのクラスにより成立している。

既に述べている様に、連鎖の成立の段階には、経緯、群化、脈絡がある。以下に各段階で定義される連鎖を概観する。

### (01)経緯で定義される連鎖

経緯で定義される連鎖には、以下のものがある。

#### 01. 意識連鎖

#### 02. 確立連鎖



### 03. 事象連鎖

### 04. 多重連鎖

実体が其の属性に属す論理原子の中から選ばれる場合, 其の実体と属性の相補関係を, 本研究では, 空と記す。空の関係で定義される連鎖が意識連鎖である。図4参照。

[0084] 空で定義される意識連鎖の属性は, 其の属性が全体の性質を有する存在である事を示す。実体が其の属性外の論理原子から選ばれる場合, 其の実体と属性の相補関係を, 本研究では, 単元と記す。単元の関係で成立する連鎖を認識連鎖と総称する。図5参照。

[0085] 確立連鎖, 事象連鎖, 多重連鎖は認識連鎖である。

[0086] 単元で定義される認識連鎖の属性は自己が部分の性質を有する存在である事を示す。

#### (02)群化で定義される連鎖

群化で定義される連鎖には, 以下のものがある。

##### 01. 群化多重連鎖

##### 02. 自然連鎖

##### 03. 群化自然連鎖

これらの連鎖も認識連鎖である。

[0087] また, 自己なる存在を成立させる候補の連鎖は, 群化の終局に定義される群化自然連鎖のひとつで, 特異連鎖(図8)と記される。

[0088] 特異連鎖は, 本章2. 8の2. 8. 2で述べる。

#### (03)脈絡で定義される連鎖

脈絡の段階では, 意識を表わす群化自然連鎖に統治されている群化自然連鎖は, 意識を表わす群化自然連鎖, 並びに意図を成立させる特異連鎖の実体を介して解き放される。

[0089] 意図を成立させる構図は図9参照。

[0090] 解き放される群化自然連鎖は, 解き放される事により, 其の性質を変える。本研究では, 性質を変えた其の群化自然連鎖を物質的存在と記す。其して, 物質的存在が

言表を成立させる。解き放される群化自然連鎖は、群化の過程を通じて統治される群化自然連鎖である。其れ故、解き放される群化自然連鎖は、群化の過程に係る群化自然連鎖が其の逆の過程を経て再定義し直されるものである。

解き放される存在(群化自然連鎖)は、意識を表わす存在を脈絡の始まりの核として、次々と存在を解き放す。其して、其れら存在もまた、核となり、次々と存在を解き放す。其して、解き放される存在が群化の始まりである多重連鎖に遭遇すれば、其の存在は次の存在を解き放す核となる事が出来ない。脈絡をモデル化して表し、其れをENWと記す。ENWは、第3章3.4で述べる。

本研究では、解き放される存在が言表可能な存在であるとし、其れらを用いて言表不可能な意識を表す存在を捉える仕組みを定義する為の論考が行われる。

#### 2.7.1 意識連鎖

意識連鎖は定義の機会に至れば、中断されない限り成立する限りの意識連鎖が定義される。意識連鎖の役割は確立連鎖から事象連鎖を成立させる触媒的役割、並びに意識連鎖の特別な形態である臨界連鎖は意図を成立させる役割を担う事である。

[0091] しかし、意識連鎖は連鎖の連糸に加わらない。

##### (01)意識連鎖の属性

存在線が成立すると、3個以上で且つ重複しない奇数個の意識原子を要素として、これが成立する限り集合が定義される。

[0092] 但し、集合間の意識原子の重複は認められる。これを意識 $\lambda$ 集合と記す。

[0093] 其の全意識 $\lambda$ 集合の組み合わせの任意のひとつが、意識 $\lambda$ 集合に順位を与える順列となる。其れを意識 $\lambda$ 集合列と記す。

[0094] 意識 $\lambda$ 集合列に属す意識 $\lambda$ 集合が意識連鎖の属性候補となる。

[0095] 意識 $\lambda$ 集合、意識 $\lambda$ 集合列を決める過程で中断が起きれば、論理原子の定義を成立させる段階に戻る。中断が起きなければ、意識連鎖の属性候補の実体を選ぶ作用に進む。

##### (02)意識連鎖の実体

意識連鎖の実体は其の属性に属す意識原子である。

- [0096] 意識連鎖の属性候補(意識 $\lambda$ 集合)に属し, 属性候補の占有速度に可能な限り小側で近似する空間速度を有する意識原子が, 意識連鎖の実体として選ばれる。実体が成立すれば其の属性候補はこの意識連鎖の属性となる。其の結果, 意識連鎖が成立する。選ばれるべき意識原子が既に別の意識連鎖の実体として選ばれていれば, 次善の意識原子が選ばれる。
- [0097] 実体を選ぶ事が出来なければ, 其の属性候補を属性とする意識連鎖は成立しない。其の場合, 意識 $\lambda$ 集合列の順番に従い次の属性候補の実体を選ぶ。この作用は意識 $\lambda$ 集合列の最後に並ぶ属性候補迄進められる。
- [0098] この過程で中断が起きれば, 論理原子を成立させる段階に戻る。
- [0099] 中断が起きなければ, 確立連鎖の実体を選ぶ作用に進む。

### (03)意識空間

意識連鎖が成立すれば, 其の実体はIDSの補集合領域に移り集合を作る。この集合を意識空間と記す。意識空間は全体を要素とする集合である。

- [0100] 回帰により, 意識連鎖の実体となる意識原子が認識原子に変われば, 其の実体は消滅する。この場合, 意識連鎖は無条件に一斉に解体される。即ち, 意識空間は消滅する。
- [0101] 転位の場合も同様の事態になる。

### 2. 7. 2 確立連鎖

確立連鎖は単元で定義される。図5参照。

- [0102] 確立連鎖は定義の機会に至れば, 1個の確立連鎖が定義される。確立連鎖の役割は連鎖の連糸の始まりを表し, 且つ意識空間に事象連鎖を成立させる為の開示を行う。

#### (01)確立連鎖の属性

存在線が成立すると3個以上で且つ重複しない奇数個の認識原子を要素とする成立する限りの集合が定義される。

- [0103] 但し, 集合間の認識原子の重複は認められる。これを認識 $\lambda$ 集合と記す。其の全認識 $\lambda$ 集合の重複順列の任意のひとつが, 認識 $\lambda$ 集合に順位を与える順列となる。其れを認識 $\lambda$ 集合列と記す。認識 $\lambda$ 集合列に属す先頭の認識 $\lambda$ 集合が確立連鎖

の属性候補となる。

- [0104] 認識 $\lambda$ 集合, 認識 $\lambda$ 集合列を決める過程で中断が起きれば, 論理原子の定義を成立させる段階に戻る。中断が起きなければ, 確立連鎖の属性候補の実体を選ぶ作用に進む。

#### (02)確立連鎖の実体

確立連鎖の実体は其の属性以外の認識原子である。即ち, 単元を成す。

- [0105] 確立連鎖の実体は, 属性候補の占有速度に可能な限り大側で近似する空間速度を有する認識原子が, 其の属性候補以外の認識原子から選ばれる。実体が成立すれば其の属性候補はこの確立連鎖の属性となる。
- [0106] 其の結果, 確立連鎖が成立する。
- [0107] 選ばれるべき認識原子が既に他の確立連鎖の実体, 或いは事象連鎖の実体として選ばれていたり, 複写回数が限界に達していれば, 次善の認識原子が選ばれる。
- [0108] 実体を選ぶ事が出来なければ, 其の属性候補を属性とする確立連鎖は成立しない。其の場合は, 論理原子を成立させる段階に戻り, 新たな時間速度の到来を待つ。
- [0109] この過程で中断が起きれば, 論理原子を成立させる段階に戻る。
- [0110] 実体が選ばれ且つ中断が起きなければ, 事象連鎖を定義する段階に進む。

#### (03)確立空間

確立連鎖が成立すれば, 其の実体はIDSの補集合領域に移り集合を作る。この集合を確立空間と記す。実体は都度蓄積される。

- [0111] 確立空間に同じ認識原子が重複して属す事は出来ない。また, 事象空間に属す認識原子とも重複する事は出来ない。
- [0112] 回帰により, 確立連鎖の実体である認識原子が意識原子に変われば, 其の実体は消滅する。この場合, 属性に属す認識原子が上記の影響を受ければ, 其の属性は解体される。其の事により其の属性の実体も消滅する。
- [0113] 転位の場合も同様の事態になる。但し, 確立空間は意識空間の様に一斉に消滅する事はない。

### 2. 7. 3 事象連鎖

事象連鎖は開示される確立連鎖に対応して, 1個定義される。其の役割は確立連

鎖に続く連鎖の連糸を成し、且つ、多重連鎖の定義を成立させる前段の役割を担う。  
図5参照。

無作為的に定義される確立連鎖は、意識空間により正式化される。其の事態が事象連鎖である。

#### (01)事象連鎖の属性

確立連鎖の定義が成立すれば、其の実体の空間速度よりも可能な限り大側で近似する空間速度を有する意識原子の実体が其れに応じる。

[0114] この関係を成立させる確立連鎖の作用を、開示と記す。

[0115] 意識空間は、其の意識連鎖の属性に属す意識原子の個数と同数であり、且つ其の意識連鎖の実体の空間速度よりも可能な限り大側で近似する占有速度を有する認識 $\lambda$ 集合を、其の確立連鎖を成立させた認識 $\lambda$ 集合列の中から選ぶ。

[0116] この関係を成立させる意識空間の作用を、連想と記す。

[0117] 連想の条件を満たす認識 $\lambda$ 集合は、1個とは限らない。其の場合は、意識空間は認識 $\lambda$ 集合列の順位の低い方の認識 $\lambda$ 集合を優先して選ぶ。

#### (02)事象連鎖の実体

事象連鎖の実体は其の属性以外の認識原子である。即ち、単元を成す。

[0118] 実体の決定方法は確立連鎖の場合とほぼ同じである。

[0119] 事象連鎖の実体は属性候補の占有速度に可能な限り大側で近似する空間速度を有する認識原子が、其の属性候補以外の認識原子から選ばれる。実体が成立すれば其の属性候補はこの事象連鎖の属性となる。

[0120] 其の結果、事象連鎖が成立する。

[0121] 選ばれるべき認識原子が既に確立連鎖の実体、或いは他の事象連鎖の実体として選ばれていたり、複写回数が限界に達していれば、次善の認識原子が選ばれる。

[0122] 実体を選ぶ事が出来なければ、其の属性候補を属性とする事象連鎖は成立しない。其の場合は、論理原子を成立させる段階に戻り、新たな時間速度の到来を待つ。この過程で中断が起きれば、論理原子を成立させる段階に戻る。

[0123] 実体が選ばれ且つ中断が起きなければ、多重連鎖を定義する段階に進む。

#### (03)事象空間



事象連鎖が成立すれば、其の実体はIDSの補集合領域に移り集合を作る。この集合を事象空間と記す。実体は都度蓄積される。

[0124] 事象空間の成立方法は確立空間の場合とほぼ同じである。

[0125] 事象空間には同じ認識原子が重複して属す事は出来ない。また、事象空間に属す認識原子とも重複する事は出来ない。

[0126] 回帰により、事象連鎖の実体である認識原子が意識原子に変われば、其の実体は消滅する。この場合、属性に属す認識原子が上記の影響を受ければ、其の属性は解体される。其の事により其の属性の実体も消滅する。

[0127] 転位の場合も同様の事態になる。但し、事象空間は意識空間の様に一齐に消滅する事はない。

#### 2. 7. 4 多重連鎖

多重連鎖は、事象連鎖が定義されれば其れに引き続き定義される。多重連鎖は事象連鎖を解す連鎖となり、且つ、群化多重連鎖の定義を成立させる前段の役割を担う。

##### (01)多重連鎖の属性

事象連鎖の定義が成立すれば、其の属性が解されて多重連鎖の属性の候補となる。即ち、事象連鎖の属性に属す認識原子の個数を $\alpha$ とすれば、解される多重連鎖の属性候補の個数は $(\alpha \alpha - 1)$ 個で表される。

##### (02)多重連鎖の実体

多重連鎖の実体は、基になる事象連鎖の実体である。実体は確立連鎖の実体、事象連鎖の実体、多重連鎖の実体と重複して、複写回数の限界迄選ぶ事が出来る。

[0128]  $(\alpha \alpha - 1)$ 個の属性候補の実体が其の途中で複写回数の限界に達すれば、其れ以降の属性候補は無役となる。実体は属性候補の占有速度が大なる方から割り当てられる。

[0129]  $(\alpha \alpha - 1)$ 個の属性候補の全てに実体が割り当てられなければ、其の属性候補を属性とする多重連鎖は成立しない。其の場合は、論理原子を成立させる段階に戻り、新たな時間速度の到来を待つ。

[0130] この過程で中断が起きれば、論理原子を成立させる段階に戻る。



[0131] 実体が選ばれ且つ中断が起きなければ、群化多重連鎖を定義する段階に進む。

[0132] 多重連鎖の実体は其の属性外の認識原子である事に於いて単元である。他方、占有速度と実体の空間時間の大小関係は、全ての多重連鎖が単元で律されているわけではない。

### (03)多重空間

多重連鎖が成立すれば、其の実体はIDSの補集合領域に移り集合を作る。この集合を多重空間と記す。実体は都度蓄積される。

[0133] 多重空間には同じ認識原子が重複して属す。また、確立空間、事象空間、に属す認識原子とも重複する事が出来る。

[0134] 回帰により、事象連鎖の実体である認識原子が意識原子に変われば、其の実体は消滅する。この場合、属性に属す認識原子が上記の影響を受ければ、其の属性は解体される。其の事により其の属性の実体も消滅する。

[0135] 転位の場合も同様の事態になる。但し、多重空間は意識空間の様に一斉に消滅する事はない。

### 2. 7. 5 群化多重連鎖

多重連鎖は、其の全てが単元で律されているわけではない。これは、連鎖を定義する律性の秩序が失なわれている事である。

[0136] この事態を改善する為に、多重連鎖並びに群化多重連鎖を群化する作用が生じる。其の作用を多重群化と記す。其して、この作用で連鎖の連糸を群化として、更に進める新たな連鎖が定義される。其の連鎖が群化多重連鎖である。

結果的に、群化多重連鎖は全て単元で律される事になる。しかし、この連鎖の属性に属す認識原子の個数が偶数個になる場合がある。この事は多重群化律で明らかになる。其の場合の連鎖は、自然連鎖となる。多重群化の作用のモデルを以下に示す。

#### (01)多重群化の骨組

01. この作用が開始される時点で多重空間に存在する実体の全連鎖、即ち、多重連鎖、並びに群化多重連鎖を要素として組合せを作り、其の任意のひとつを多重連鎖順序列とする。

[1, 2, 3, …]

02. 多重連鎖順序列を基に成立する限りの連鎖の対が、以下の様に作られる。この連鎖の対を多重群化対、其の順序列を多重群化対の順序列と記す。

[(1, 2), (1, 3), (1, 4)…, (2, 3), (2, 4), (2, 5)…, (3, 4), …]

03. 多重群化対(X, Y)で、Xの実体の空間速度がYの実体の空間速度よりも大となれば、この関係を多重群化の主従対と記す。

04. この主従対が成立すれば、Xを主、Yを従とする多重群化律が実行される。この主従対が成立しなければ其の多重群化対は無役となり、次の主従対に多重群化律が進められる。

05. 多重群化律は、主従対の成立順番に従い無役の多重群化対を飛び越し実行される。

06. 多重群化律は連鎖X, Yから新たな連鎖Zの属性を定義する作用を行う。

07. 群化多重連鎖の実体は単元で定義される。

08. この過程で中断が起きれば、論理原子を成立させる段階に戻る。

09. 多重群化律が全ての主従対に作用し終われば、論理原子を成立させる段階に戻る。

10. 多重群化で自然連鎖が成立すれば、引き継がれる多重群化の継続点を残し、自然群化に進む。

11. 自然群化で群化多重連鎖が成立すれば、多重主従対の成立順番の継続点に戻る。

## (02)多重群化律

01. 連鎖Xの属性に属す認識原子の空間速度を大小の順に並べる。

02. 連鎖Yの属性に属す認識原子の空間速度を大小の順に並べる。

03. 上記01, 02の認識原子の順番を対応付ける。

04. 対応付けられる認識原子の空間速度が $X > Y$ となる場合に限り、XとYの属性の認識原子を新たな連鎖(群化多重連鎖)の属性の要素として取り込む。

05. 対応付けられないXの認識原子は、新たな群化多重連鎖の属性候補として取り込む。

06. 多重群化律ではXとYから新たな連鎖(群化多重連鎖)の属性を定義する為の作用が行われる。

07. 新たな群化多重連鎖の属性候補の認識原子の空間速度を大小の順に並べる。

08. 同じ空間速度の認識原子は其の後位の方から除く。

[0137] ここで、属性候補の認識原子の個数が偶数であれば、其の属性候補は群化自然連鎖の属性候補となる。其して、属性候補の認識原子の個数が奇数であれば、其の属性候補は群化多重連鎖の属性候補となる。

09. 多重空間は、この群化多重連鎖の属性候補が属性となる為の実体を単元で選ぶ。選ばれる認識原子が複写の限界を越えていなければ、其の認識原子は実体となる。其して、其の属性候補は属性になる。この場合、其の属性に属す認識原子の個数が偶数となれば、其の連鎖は群化自然連鎖、奇数となれば、其の連鎖は群化多重連鎖となる。選ばれる認識原子が複写の限界を越えていれば、次善の認識原子が選ばれる。実体を選べなければ、この属性候補は無役となる。

10. 群化多重連鎖の実体は多重連鎖の実体と同様に多重空間の要素になる。

#### 2. 7. 6 自然連鎖

多重群化律で成立する群化多重連鎖の属性に属す認識原子の個数が偶数となる場合、この連鎖が成立すると其の多重群化の作用は停止され、この連鎖の実体は多重空間と異なる空間に移される。この連鎖を自然連鎖、其の異なる空間を自然空間と記す。

自然連鎖が成立すると多重連鎖の群化律とは異なる群化律が成立する。其れを自然群化律と記す。これは自然連鎖の属性に属す認識原子の個数が偶数である事により生じる作用である。

[0138] この自然群化律で定義される連鎖を群化自然連鎖と記す。

[0139] 自然群化律は、本章2. 7の2. 7. 7で述べる。群化自然連鎖の実体は、自然空間の要素になる。

#### 2. 7. 7 群化自然連鎖

自然連鎖の属性に属す認識原子の個数が偶数になる事において、其の事態を改善する為に、多重群化律に準じる作用が生じる。其して、この作用で連鎖の連糸を更

に進める新たな連鎖が定義される。其の連鎖が群化自然連鎖である。

[0140] この場合にも、この連鎖の属性に属す認識原子の個数が奇数個になる場合がある。この事は本章2.7の2.7.7で述べる自然群化律で明らかになる。

[0141] 自然群化の作用のモデルを以下に示す。

(01)自然群化の骨組

01. この作用が開始される時点で自然空間に存在する実体の全連鎖、即ち、自然連鎖、並びに群化自然連鎖を要素として組合せを作り、其の任意のひとつを自然連鎖順序列とする。

[1, 2, 3, …]

02. 自然連鎖順序列を基に成立する限りの連鎖の対が以下の様に作られる。この連鎖の対を自然群化対、其の順序列を自然群化対の順序列と記す。自然群化対は多重群化の場合と若干異なる。

[(1, 2), (1, 3), (1, 4)…, (2, 3), (2, 4), (2, 5)…, (3, 4), …]

03. 無

04. 自然群化対では多重群化対の場合の様に主従対の条件は不要である。Xを主、Yを従とする自然群化律が実行される。

05. 自然群化律は自然群化対の成立順番に従い実行される。

06. 自然群化律ではXとYから新たな連鎖(群化自然連鎖)の属性を定義する為の作用が行われる。

07. 群化自然連鎖は単元で定義される。

08. この過程で中断が起きれば、論理原子を成立させる段階に戻る。

09. 自然群化律が全ての主従対に作用し終われば、論理原子を成立させる段階に戻る。

10. 自然群化で多重連鎖が成立すれば、引き継がれる自然群化の継続点を残し多重群化の継続点に進む。

11. 無

(02)自然群化律

01. 連鎖Xの属性に属す認識原子の空間速度を大小の順に並べる。

02. 連鎖Yの属性に属す認識原子の空間速度を大小の順に並べる。
03. 上記01, 02の認識原子の順番を対応付ける。
04. 対応付けられる認識原子の空間速度が $X > Y$ となる場合に限り, XとYの属性の認識原子を新たな連鎖(群化多重連鎖)の属性の要素として取り込む。
05. 対応付けられないXの認識原子は, 新たな群化自然連鎖の属性候補として取り込む。
06. 対応付けられないYの認識原子は, 新たな群化自然連鎖の属性候補として取り込まない。
07. 新たな群化自然連鎖の属性候補の認識原子の空間速度を大小の順に並べる。
08. 同じ空間速度の認識原子は其の後位の方から除く。ここで, 属性候補の認識原子の個数が奇数であれば, 其の属性候補は群化多重連鎖の属性候補となる。

[0142] 其して, 属性候補の認識原子の個数が偶数であれば, 其の属性候補は群化自然連鎖の属性候補となる。

09. この群化自然連鎖の属性候補を満たす実体を単元で選ぶ。選ばれる認識原子が複写の限界を越えていなければ, 実体となる。其して, この属性候補は属性になり, 属性に属す認識原子の奇数偶数により, 群化自然連鎖, 群化多重連鎖が成立する。

[0143] 選ばれる認識原子が複写の限界を越えていれば, 次善の認識原子が選ばれる。実体が選べなければ, この属性候補は無役となる。

10. 群化自然連鎖の実体は自然連鎖の実体と同様に自然空間の要素になる。

## 2. 7. 8 仮想存在の属性に関する仮想の為の思想

ここでは, 仮想存在(連鎖)を決定付ける属性がどのような思想を基にして論考されているかを告知する。

(01)本研究が仮想する存在の属性は集合で表される。

(02)本研究では其の集合の要素の概念は空間と時間で定義されるものとする。仮説される論理原子はこの定義を具現化する為のものである。

(03)本研究では存在を空間に割り込む空間と定義する。そして, 其の空間は時間の概念を帯同し, 其の影響を受けるものとする。



(04)これは存在の非絶対性を宣言する為である。本研究で用いられる思想である。

(05)論理原子は存在の属性を定義するだけに止まらず、存在の実体の概念を定義する事が出来る様に仮説されている。

(06)本研究の存在の属性の集合に属す要素(論理原子)の個数は奇数個である事に其の本質性があるものとする。

(07)本研究で定義される属性の候補である意識 $\lambda$ 集合, 認識 $\lambda$ 集合に属す論理原子の個数は奇数で定義されるのはこの本質性の為である。

(08)境界原子(論理原子)はこの本質性を意義付ける為の基本的な概念として定義されている。

(09)論理原子は存在の属性, 存在の実体, そして, 境界原子の概念の定義が可能となる様に仮説されている。

(10)確立連鎖, 事象連鎖の属性は其の本質性を満たしている。

(11)意識連鎖の属性は其の本質性を満たしている。

(12)本研究の課題は自己なる存在を定義する事にある。

(13)其の発端として, 確立連鎖が誕生する。

(14)意識連鎖も確立連鎖と同じ様に誕生する。

(15)意識連鎖は可能的に成立する限り定義されるが, 確立連鎖は其の誕生周期に於いては唯1個である。

(16)其の確立連鎖は誕生すると意識連鎖により, 正規化される。

(17)誕生する1個の確立連鎖と既に成立している複数の意識連鎖の間で開示が行われ, 其の確立連鎖が成立する周期では, 其れに対応する意識連鎖が連想して, 1個の事象連鎖が誕生する。

(18)本研究の仮説では存在の発端である確立連鎖は確率的な作用により定義されるものとする。他方, 存在は何らかの規範で律されているものと期待する。

[0144] 本研究に於ける開示, 連想の仕組は確率的な存在を規範に基づく存在に定義し直す為の作用であり, 其れはこの思想を反映させる為である。

(19)本研究では仮想される存在の属性候補となる論理原子の集合の列がこの規範を与える縁として位置付けられる。



(20)本研究の課題は「自己なる存在」を定義する事である。其の過程を本研究では存在の連糸と呼んでいる。

(21)確立連鎖が誕生し、意識連鎖により規範が与えられ事象連鎖が誕生する。

[0145] 本研究では事象連鎖は、意識連鎖により正規化され、最早、確率的な存在ではないと考える。

(22)他方、事象連鎖が成立しても、其れで「自己なる存在」が定義出来るわけではない。其の為には更に連糸を紡ぐ必要がある。

(23)更に連糸を紡ぐ為には更に新たな作用を求め、其れにより属性を定義し、新たな存在を成立させる事である。この段階では事象連鎖の属性から新たな作用で属性を定義し、事象連鎖と異なる連鎖を成立させる事である。

(24)事象連鎖の属性から新たな属性を定義するとは其の属性である集合の要素(認識原子)を多重的に再定義し直し、集合を定義する事である。この段階では、これ以外の作用を見つける事が出来ないからである。

[0146] 其の集合を属性とする連鎖が多重連鎖である。

[0147] しかし、多重連鎖が成立しても、其れで「自己なる存在」が定義出来るわけではない。其の為には更に連糸を紡ぐ必要がある。

(25)更に連糸を紡ぐ為には更に新たな作用を求め、其れにより属性を定義し、新たな存在を成立させる必要がある。

[0148] この段階では、多重連鎖の属性から新たな作用で属性を定義し直し、多重連鎖とは異なる連鎖を成立させる事である。

(26)多重連鎖の属性から新たな属性を定義するとは其の属性である集合間の要素(認識原子)を可能な限り組み合わせ、新たな集合を定義し直す事である。この段階では、これ以外の作用を見つける事は出来ないからである。

[0149] 其の様にして再定義し直される集合を属性とする連鎖が群化多重連鎖である。

(27)多重連鎖並びに群化多重連鎖の属性の集合間の要素を組み合わせ、新たな属性を再定義する作用の律性が多重群化律である。

[0150] しかし、群化多重連鎖が成立しても、其の事で「自己なる存在」が定義出来るわけではない。其の為には更に連糸を紡ぐ必要がある。

(28)この群化多重連鎖の属性を定義する過程で、其の集合に属す要素(認識原子)の個数が偶数になる事がある。

[0151] 其の様な集合を属性とする連鎖は群化多重連鎖ではあり得ない。其の連鎖が自然連鎖である。

(29)しかし、この自然連鎖が成立しても「自己なる存在」が定義出来るわけではない。其の為、この自然連鎖から更に連糸が紡がれる。

(30)多重群化律が生じる理由は多重連鎖の実体が重複する事、並びに其の属性に属す認識原子が重複する事である。そして、自然連鎖が生じる理由は自然連鎖の実体が重複する事、並びに其の属性に属す認識原子が偶数となる事である。

(31)上述の理由から自然連鎖が誕生すると、其れ迄の多重群化律は停止し、自然連鎖の間で群化作用が生じる。其の作用が自然群化律である。

(32)自然群化律は多重群化律に準じる作用で、自然連鎖並びに群化自然連鎖の属性の集合間の要素を多重群化よりも更に可能な限り組み合わせる新たな属性を再定義する作用である。この連鎖が群化自然連鎖である。

(33)自然群化律で属性に属す要素(認識原子)の個数が奇数になれば、其の群化自然連鎖は群化多重連鎖となる。

[0152] 其の事により、自然群化律は停止し、多重群化律が再開される。即ち、多重群化律と自然群化律は其処で決定される属性により其れに応じる連鎖を成立し、連糸を紡ぐ。

(34)本研究では多重群化律並びに自然群化律が仮説される。

(35)群化自然連鎖は其の群化の終局で特異連鎖に到達する。其の事により、臨界連鎖により「自分なる存在」の定義が可能になる。

(36)「自己なる存在」には意識を表す連鎖を決定し、其処から脈絡を構成する存在が出現する。其の脈絡の展開される様相が、本研究で言う思考作用である。

(37)本研究では所謂自然界に於ける森羅万象は全て「自己なる存在」を成し、其処では思考作用が行われていると考えるのである。

## 2.8 自己なる存在のモデル

自己なる存在のモデルは、図11で示す。

[0153] 自己とは決して成立する事のない全体と部分が等価になる関係が成立する事である。これが、本研究で論考される意図の構図(図9)である。本研究で言う自己なる存在とは、この意図を基に成立する構図の事である。

[0154] 其して、この関係に於ける全体(臨界連鎖)は、其の部分(特異連鎖)の中に其の全体(臨界連鎖)と等価になるより妥当的な部分(群化自然連鎖)を探索する。この探索は、最初に全体(臨界連鎖)に対応する部分(特異連鎖)が境界原子を実体とする疑似性を有する事から、其れを取り除く為に行われる作用である。探索される部分(群化自然連鎖)が本研究で論考される意識を表す存在となる。

探索される意識を表す存在(群化自然連鎖)は、其の存在が統治する存在(群化自然連鎖、含む自然連鎖)を客体化と陳述により解き放す。

[0155] 解き放す存在は脈絡を成し、意識を表す存在が全体に対応する部分である事を言表する。この過程を、本研究では思考のモデルと考える。

解き放される存在は、自己なる存在の特異連鎖の実体の中で成立する関係である。其の事により、解放される存在は自己なる存在の中で成立する。

[0156] 換言すれば、解き放される存在はそれぞれが新たな自己なる存在を成立させる事を意味する。

[0157] 本研究では、森羅万象は全て自己なる存在となる。其して、其のひとつひとつが自己となる。

記述の様に、自己なる存在を代表するのは1個の特異連鎖である。其の特異連鎖には、複数の臨界連鎖が呼応する。其の事により、1個の特異連鎖には複数の意識を表す存在が成立する事になる。

以上の事から、自己なる存在は、(01)特異連鎖、(02)臨界連鎖、(03)意識を成立させる連鎖で構成される。自己なる存在が、意識を表す存在が統治する存在を解き放す。

[0158] この作用が連糸の最終局面である。

## 2. 8. 1 臨界連鎖

意識連鎖の属性候補が境界原子を実体として選ぶ事により、成立する意識連鎖を

臨界連鎖と記す。臨界連鎖とは、境界連鎖を実体とする意識連鎖の事である。

[0159] 意識連鎖を定義する仕組は、属性の占有速度に内側から近似する意識原子を実体として選ぶ事である。即ち、空の関係を成立させる事である。

[0160] 其れ故、属性に境界原子が含まれる場合、臨界連鎖は成立する。図7にこのモデルを示す。臨界連鎖は全体の性質の極致を表す。

### 2. 8. 2 特異連鎖

群化自然連鎖の属性候補が境界原子を実体として選ぶ事により成立する群化自然連鎖を特異連鎖と記す。特異連鎖とは、境界連鎖を実体とする群化自然連鎖の事である。

[0161] 群化自然連鎖を定義する仕組は、属性の占有速度に外側から近似する認識原子を実体として選ぶ事である。即ち、単元の関係を成立させる事である。

[0162] 通常では、境界原子が群化自然連鎖の実体になる事はない。しかし、複写の限界から、実体として選ばれる認識原子が次善に選ばれる認識原子も含めて、最早、選ばれる事が出来なくなれば、境界原子が選ばれる。

境界原子は意識原子に属す論理原子であるが、存在線上意識原子と認識原子の境界に位置し、両者を兼ねる性質を有する論理原子である。其の事が境界原子が群化自然連鎖の実体になり得る理由である。

[0163] その場合の群化自然連鎖が特異連鎖である。付言すれば、特異連鎖とは究極の群化自然連鎖の事である。図8にこのモデルを示す。特異連鎖は部分の性質の極致を表す。

### 2. 8. 3 意図のモデル

臨界連鎖の属性は他の意識連鎖と同様に空で其の実体を選ぶので、他の意識連鎖と同様に其の実体を介して其の属性が全体である事を叙述する。

[0164] 他方、特異連鎖の属性は他の群化自然連鎖と同様に単元で其の実体を選ぶので、他の群化自然連鎖と同様に其の実体を介して其の属性が部分である事を叙述する。臨界連鎖、並びに特異連鎖の実体は、共に境界原子である。其れ故、この状態は全体が一致する関係を表す事になる。本研究ではこの関係を意図として定義する。

この関係は、特異連鎖が成立する事により、既に成立している臨界連鎖が其の特

異連鎖に呼応して成立する。其の呼応関係は、特異連鎖1個に対し、臨界連鎖は $n$ 個となる。即ち、1個の特異連鎖が成立すれば、其の特異連鎖に統治される群化自然連鎖で其の属性に属す認識原子の数と同じ数の意識原子を属性とする臨界連鎖が其の特異連鎖に呼応する。其して、其の群化自然連鎖が意識を表す存在となる。即ち、1個の特異連鎖には最大臨界連鎖の数の意識を表す存在が成立する関係になる。図9にこの関係を示す。

#### 2. 8. 4 意識のモデル

特異連鎖の属性に属す認識原子の個数は、認識原子の総数となる事に於いて其の特定は不可能になる。

[0165] 他方、臨界連鎖の属性に属す意識原子の個数は総数ではない。

[0166] 其れ故、其の数を $m$ 個とすれば、特異連鎖に至る群化の過程で $m$ 個の認識原子を属性とする群化自然連鎖が成立している。この関係を成立させる群化自然連鎖が意識を表す存在である。

[0167] この関係は、意図の成立の基で定義される。同じ個数の認識原子を要素とする群化自然連鎖が複数存在する場合には、実体の空間速度の大なる方が優先して選ばれる。図10にこの関係を示す。

#### 2. 9 思考モデル

意識を表す連鎖が決まると、其の連鎖の実体並びに意図を成立させる特異連鎖の実体を介す事により其の意識を表す連鎖の属性が新たな自己なる存在を言表する。

[0168] この存在が脈絡で定義される存在である。

[0169] 存在の連糸は、この存在により更に進められる。本研究では、この存在が進める連糸がわれわれの思考に相当する作用であると考え。其の意味で、脈絡の構造は自己なる存在に成立する思考モデルとして位置付ける事が出来る。

脈絡の過程は、多重連鎖を起因として始まる自然連鎖の群化の過程が逆に辿られる様相の事である。図6を再見すれば、この事が確認される。

群化の段階では自己なる存在が未だ成立していない状態で存在が作り出される。其れに対し、脈絡の段階では自己なる存在が更なる存在を作り出す関係になる。



[0170] これが、群化と脈絡が作り出す存在の違いである。

#### 2. 9. 1 脈絡の定義

自己なる存在が成立すれば、其の基で意識を成立させる群化自然連鎖が選ばれ、其の存在が核となり、其の存在が統治している群化自然連鎖(複数)を其の意識を表す存在の実体、並びに、特異連鎖の実体を介して言表される。この存在が解き放される存在である。選ばれる群化自然連鎖が統治している群化自然連鎖は、自然連鎖順序列(本章2. 7. 7(01))に属しているものである。この構造を図12で示す。

解き放される存在は、客体化、並びに陳述として区別される。陳述で定義される存在には記憶を成立させる存在と記憶を成立させられない存在とが生じる。記憶を成立させられない存在を同化される存在と記す。これらの定義は、本章2. 9. 4と2. 9. 5で述べる。但し、思考モデルでは同化される存在は除かれる。

客体化、または陳述により群化自然連鎖列の中から選ばれる群化自然連鎖は、重複して選ばれる事はない。其れ故、選ばれる群化自然連鎖がなくなるとは、其の次に選ばれる連鎖が多重連鎖となる場合に相当する。

[0171] 何故なら、既に述べた様に脈絡は群化の過程が逆に辿られる関係を示すからである。

[0172] 其して、其の群化が行われる契機は、多重連鎖となるからである。

脈絡の様相をわれわれの言葉に対応付ければ、われわれが発報する最初の言葉は客体化される存在に相当する。其れに引き続き発報される言葉は、陳述される存在に相当する。

意識を表す存在、客体化を成立させる作用、並びに陳述を成立させる作用を言表する事は不可能である。即ち、われわれが発報する言葉の直前の状態とは、意識を表す存在、客体化を成立させる作用、陳述を成立させる作用に他ならない。其れらは全て言表不可能である。

ここで、ソフトを開発する行為を考察する。

[0173] 其の行為は、例えば以下の様な問いに対する答えを求める為の思考作用に相当する。即ち、林檎とは何なのか。認識とは何なのか。林檎が時と共に変容する原因は何なのか。林檎は何故変容するのか。誰が林檎の変容を認識するのか。言葉とは何



なのか。われわれが言葉を発する原因は何なのか。われわれは何を考えるのか。林檎は考えるか。物質とは何なのか。物質でない存在とは何なのか。感情とは何なのか。意識とは何なのか。自己は何を意味するか。意味とは何なのか。われわれが本を読む原因は何なのか。われわれが文章を作る原因は何なのか。われわれが言葉を聞く原因は何なのか。われわれが食べる原因は何なのか。われわれは何故生まれるか。われわれは何故動くか。われわれは何故事故に遭うか。われわれは何故死ぬか。われわれは何故喜ぶか。われわれは何故悲しむか。われわれは何故群れるか。われわれとは何なのか。

[0174] 其して、存在とは何なのか。

仮に、開発するソフトが日常的で簡単なプログラムである場合でも、其れを開発する際のわれわれの思考行為は上記の問の答えを探す作業に等しくなる。付言すれば、ソフトを開発する思考の背景には、深遠な哲学的課題が潜むと考える。

換言すれば、ソフトの本質性は、経験や成果の蓄積が効果を齎す、例えば工学的世界とは異なる世界を形成している。其れ故、これまでの科学的方法論で、ソフトの本質に迫る事は不可能であると思われる。この点に留意すれば、今、ソフトに求められる事は、機能的効果に留まらずソフトを成立させる原理、並びに理論である。これらは、空想の世界に其の始まりを置き、例えば、発見的論理学の認識世界を再構築する事により得られる成果である様に思われる。

## 2. 9. 2 客体化

自己を表す存在は、意識を表す存在の属性に属す認識原子を実体とする群化自然連鎖を、群化自然連鎖列の中から選ぶ。

[0175] 本研究では、この選ばれる群化自然連鎖を客体化される存在と記す。

[0176] この過程を介する事により、群化自然連鎖列に属す言表不可能な存在(群化自然連鎖)が言表可能な存在に変わる。言表可能な存在の定義は、本章1. 2で述べられている。

## 2. 9. 3 陳述

自己を表す存在は、客体化される存在の属性に属す認識原子を実体とする群化自然連鎖を、群化自然連鎖列の中から選ぶ。但し、自己を表す存在は、既に選ばれて

いる群化自然連鎖を繰り返し選ぶ事が出来ない。

[0177] 本研究では、この選ばれる群化自然連鎖を陳述される存在と記す。

[0178] 陳述で選ばれる群化自然連鎖は、客体化される存在と同じ言表可能な存在となる。

[0179] 陳述の作用は、群化自然連鎖列の中から選ばれる群化自然連鎖がなくなるまで継続される。群化自然連鎖列は、群化の項で述べられている。

#### 2. 9. 4 同化

客体化される存在、並びに陳述される存在の中で、属性に属す全ての認識原子が存在の実体となっている場合、本研究では、其の様な客体化される存在、並びに陳述される存在を、同化が成立する存在と記す。

[0180] 同化の成立する存在は、言表可能な存在でありながら、言表不可能な存在となる。例えば、本研究では、われわれが無意識に行う振る舞いは、この存在の成立を意味する。

#### 2. 9. 5 記憶

客体化される存在、並びに陳述される存在の中で、属性に属す全ての認識原子が存在の実体となり得ない場合、本研究では、其の様な客体化される存在、並びに陳述される存在を、記憶が成立する存在と記す。

[0181] 其の存在は、言表可能な存在となる。

### 第3章 仮想空間(TDM)

自己の意識に内包されている存在が自己を介して解き放されるとは、新たな存在が自己の内に成立する事である。本研究では、この過程を脈絡と呼び、自己なる存在の思考過程を表すものとする。

[0182] 本研究では、この脈絡の過程において特に意識を表す存在を言表する事が出来ない事がわれわれの思考における曖昧さを生じさせる原因になっていると考える。

本研究が目指す所は、言表不可能な存在を用いて意識を成立させる存在を捉える事にある。其の事により、思考における曖昧さを取り除く仕組みを求めようとするものである。

[0183] 本章では、この仕組のモデル(TDM)を論考する。

### 3. 1 存在の位置関係

脈絡上、自己なる存在は、過去、今、未来の自己に分けられる。其して、今の存在を自己とすれば、過去の存在は過去の自己を意味し、其れを祖先、未来の存在は未来の自己を意味し、其れを子孫と位置付ける。

[0184] 脈絡の基で、今の自己は以下の様に仕分けされる。

(01)客体化される今の自己の祖先は不明となる。

(02)陳述される今の自己には祖先、並びに子孫がいる。

(03)最後の陳述の今の自己には、子孫がない。

本来、意識を表す存在の言表が成立させられれば、思考の曖昧さを乗り越えられる。しかし、其の存在は言表が不可能である。

[0185] 解き放される存在は言表可能で、其れらは1個の意識を表す存在に端を発している。

[0186] 本研究では、其れら存在の言表の構造を求める。其の構造により、解き放される存在に言表を成立させる。其して、其の言表の集合と言表不可能な意識を表す存在を呼応させるモデルが論考される。

### 3. 2 言表の仕組

本章3. 1の今の自己を言表する仕組を述語構造(PS)と記す。

[0187] 其の為に連鎖の役割が用いられる。即ち、

(01)過去でも未来でもない自己を用いて、過去でも未来でもない今の自己は、確立連鎖の役割を用いて言表することが出来る。

(02)未来の自己を用いて言表される今の自己は、事象連鎖の役割を用いて言表する事が出来る。

(03)過去の自己を用いて言表される今の自己は、多重連鎖の役割を用いて言表する事が出来る。

### 3. 3 言表の構造

ここでは、PSの構造を述べる。

[0188] PSの構造は図14, 15, 16, 17で見る事が出来る。

[0189] 其の構造は、7種の叙述と其の4種の叙述を記憶する専用領域で定義される。

[0190] 存在とは、自己で自己を定義する叙述の事である。PSもまた叙述で定義される。しかし、存在を定義する叙述と異なるのは、PSの叙述は其の構造により、言表に換わる事である。

叙述が言表に換わる為には第2章2. 2の言表で述べている様に、叙述は外延的に成立するので、叙述する存在は其れを成立させる為には何かに内包されている事が必要である。PSの構造は、自己に係わる二つの存在を自己を内包する存在とする。自己に係わる二つの存在とは、自己の祖先と自己の子孫である。

因に、祖先は自己に内包される存在である。しかし、自己が其の叙述を成立させる為には祖先を外延的に置き直す必要がある。其の事は、ENWの項目で述べる。

PSに於ける4種の領域は、この二つの存在が自己を内包化する役割を果たす為に設けられている。其の事により、PSの叙述が成立している。其れ故、PSにおけるこの領域の役割は重要である。

### 3. 4 脈絡構造の再定義

図12で明らかな様に、今の自己の子孫は今の自己の外延的な存在である。其れ故、今の自己はPSを用いて其の子孫を叙述する事が出来る。

[0191] 他方、自己の祖先は今自己に内包される関係になる。其れ故、今の自己はPSを用いて自己の祖先を叙述する事が出来ない。

[0192] この問題を克服する為に、図12の脈絡の構造は過去の自己が今の自己の外延的な位置関係を成す様に書き改められる。

[0193] 書き直される結果、脈絡上の存在は今の自己、或いは未来の自己に位置付けられる以外の存在は、例えば鏡に映る存在の様に虚像化される。即ち、脈絡上における存在が祖先となる場合、其の存在は虚像化される。其の構造が図18で示される。

ENWでは、今の自己の存在は以下の様に証明される。即ち、  
(01)過去でも未来でもない自己の存在は、PS2(図15)で証明される。

(02)未来の自己の存在は、PS3(図16)で証明される。

(03)過去の自己の存在は、PS4(図14)で証明される。

(04)未来を有しない自己の存在は、PS2-1(図17)で証明される。

ここで、上記(03)を成立させる為には、ENWの過去に位置する存在を未来に置き換える事が必要である。何故なら、言表は外延的に成立するので、過去を言表する事が出来ないからである。

[0194] 其の結果、再定義される脈絡の過去の自己は脈絡に属す自己とは異なり、虚像化される自己に置き換えられる。

### 3. 5 存在の証明

存在の証明とは、存在の言表を成立させる仕組みを知り、其の仕組みで存在を叙述する事である。存在は、属性と主語の相補関係で定義される。これが存在を定義するための叙述である。この叙述は、既に述べている様に自己で自己を定義するに留まり、其れが其の存在の成立を証明している訳ではない。

脈絡上では、記述の様に今の自己と其の祖先、並びに子孫の相対的な関係が成立している。この関係を用いて、今の自己の存在の証明を行う事が出来る。即ち、今の自己の存在の証明は、自己の祖先と未来の子孫の存在の証明が成立すれば、結果的に今の自己が成立している事になる。

存在が存在する事を証明する為に係る不可避的な事情を以下に示す。

(01)存在とは自己を定義する叙述の事であるが、其の叙述は其の存在が存在する事を証明する役割は果たし得ていない。

(02)言表とは、存在が存在する事を証明する叙述の事である。

(03)言表、並びに叙述は、外延的に成立する。

(04)言表は、記憶が成立する存在に成立する。本研究では、記憶が成立する存在を物質的存在と考える。

(05)存在の言表は、自己を表す存在(図11)の基で成立する。

本研究では、この叙述を成立させる構造をPSと記す。換言すれば、PSが叙述されれば其れが言表となる。其して、其のPSは存在の存在を証明している事になる。



存在が成立している事を証明する為には、以下の事柄が特定されなければならない。即ち、

- (01)証明されるべき存在を特定する事
- (02)PSの構造を特定する事
- (03)PSを成立させる叙述を特定する事

上記(01)の存在は、解き放される存在の脈絡から特定する事が出来る。第4章では、この存在は主語として定義される。上記(02)のPSの構造は、存在の構造、叙述の性質、其して証明の意義を基に定義される。上記(03)の叙述は、脈絡上の存在の位置的な相対関係から定義される。

図13で2として示される存在は、本章3. 1で示される(01)の事である。この場合、其の存在の子孫は不明ではないが祖先は不明である。この位置関係にある存在、即ち、客体化される存在が成立する事を証明する言表の構造は、確立連鎖の叙述を用いてPSを成立させる事である。本研究では、其れをPS2と記す。

[0195] 図13で6として示される存在は、本章3. 1で示される(03)の事である。この場合、其の存在の祖先は不明ではないが子孫は不明である。この位置関係にある存在は、陳述の終端に位置する存在である。この存在が成立する言表の構造は、確立連鎖の叙述を逆に換えてPSを成立させる事である。本研究では、其れをPS2-1と記す。

図13で3-5として示される存在は、本章3. 1で示される(02)の事である。この場合、其の存在の子孫、並びに祖先は不明ではない。この位置関係にある存在は、客体化される存在、陳述の終端に位置する存在を除く存在である。これら存在が成立する事を証明する言表の構造は、子孫については事象連鎖の叙述、祖先については多重連鎖の叙述を用いてPSを成立させる事である。本研究では、其れをPS3, PS4と記す。

PS2-1の為の叙述、即ち、確立連鎖の叙述を逆に換える叙述を成立させる連鎖は、存在していない。其の為、本研究ではPS4の叙述を否定し、且つ、PS2の叙述を可逆させる事により、其の存在の言表を成立させる。PS4の叙述を否定する叙述をPS4N, PS2の叙述を可逆させる叙述をPS2Nで表す。この件については、図17参照。図13は、以上のことが要約して示されている。

ここでは、PSを成立させる叙述の特定について述べる。

- [0196] 叙述は、脈絡上の存在の位置的な相対関係から次の様に求める。脈絡上の存在の位置的な相対関係は、以下の様に(01)(02)(03)に分けられる。本研究では、この相対関係を用いてPSを成立させる叙述を特定する。

(01)客体化される存在

この場合の存在の相対的な関係は、不完全である。即ち、過去の自己は意識を表す存在となり言表不可能、其して、未来の自己は言表可能である。

- [0197] この場合、今の自己を叙述出来るのは確立連鎖の叙述である。其の確立連鎖の叙述で、今の自分をPS2として成立させれば、其のPS2はこの場合の今の自己を言表する必要且つ十分条件となる。其のPSを図15(PS2)で示す。

- [0198] 因に、其のPSの存在は第4章では名詞の1種別である正規(表2)が該当し、L2で叙述されるベクトルに換わる。其の正規の狭義の属性は、入力である。

(02)陳述による存在

この場合の存在の相対的な関係は、言表可能な過去の自己、並びに、未来の自己として成立する。この場合、今の自己が過去の自己を叙述出来るのは事象連鎖の叙述である。其の事象連鎖の叙述で、今の自己をPS3として成立させれば、其のPS3は過去の自己を言表する十分条件となる。

- [0199] 同じく、今の自己が未来の自己を叙述出来るのは、多重連鎖の叙述である。其の多重連鎖の叙述で、今の自己をPS4として成立させれば、其のPS4は未来の自己を言表する必要条件となる。即ち、この場合の今の自己の成立は、PS3並びにPS4の言表で証明される。PS3を図16、PS4を図14で示す。

- [0200] 因に、其のPSの存在は第4章では名詞の1種別である正規(表2)が該当し、L3、L4で叙述されるベクトルに換わる。其の正規の狭義の属性は、出力である。

第4章で説明される名詞の種別(表2)には、Kと記される名詞がある。Kは、PS3、PS4、並びにPS2-1で叙述される。

(03)未来の自己が言表不可能となる存在

この場合の存在の相対的な関係は、不完全である。即ち、過去の自己は言表可能、しかし、未来の自己は言表不可能である。この事態は陳述の最後に現れる構図で

ある。この場合、今の自己を叙述出来るのは、確立連鎖の叙述を可逆させる叙述である。これは、PS2の言表がPS4で否定される構造として定義される。

- [0201] 其れ故、この証明は修正されるふたつのPS、即ち、PS4N、PS2Nで言表される。其れをPS2-1として図17で示す。PS2-1、即ち、PS4N、並びにPS2NはそれぞれPS4、並びにPS2と同族に扱われる。

PS4の第2規約では「多重連鎖の叙述を用いて未来の自己」が叙述されるが、PS4Nの第2規約では「事象連鎖(過去)の叙述を多重連鎖(未来)の叙述に置き換えて未来の自己」が叙述される。これは、PS4の第2規約が叙述する未来の自己を否定する叙述を意味する。

- [0202] 更に、PS4Nの第4規約では「未来の自己の成立を宣言し、且つ、PS2Nの第4領域にこの宣言を移す」。其して、この場合、PS2の叙述は図17で示されるPS2Nの様になる。PS4Nの基で定義されるPS2Nが、PS2を可逆させる意味を表す。

因に、PS4N、並びにPS2Nで言表される存在は、第4章では名詞の1種別であるM(表2)が該当する。PS2-1は、PS2、PS4が修正されるPS2N、PS4Nで定義される。因に、このPS2N、PS4Nで言表される存在は、第4章ではL2、L4で叙述されるベクトルに換わる。

### 3.6 TDMの定義

図18をモデル化し、其れをTDMと記す。本研究では、ENWの元になる脈絡は特異連鎖の実体において成立するので、ENW、並びにTDMも其の様に位置付ける。

- [0203] 脈絡に属す存在はPSにより3種族に仕分されるので、ENW、並びにTDMにおいても同じように3種族に仕分けされる。

PSの3種族は集合を成す。概念的に其れらは3種の座標軸となる。其の座標軸の役割とは言表を成立させる事であり、其れは、其の座標軸に属すPSが相補する事により成立するものである。

- [0204] 其の相補作用は、それぞれの座標軸が回転する事に譬えられる。其して、この座標軸の回転は順序的に行われ、且つ、其の順序がひと通り終了する都度、3種の座標軸全体が1回転し、其れを周期として成立する。

[0205] この一連の作用を繰り返す事によりPSが言表を成立させる。其の結果、TDMの疑似空間に意識を表す存在が形成される関係に譬える事が出来る。図19にこのモデルを示す。

脈絡に準じれば、PSはPS2→PS3→PS4→PS2-1の順に成立している。

他方、本研究の目的から、脈絡の構造はENWに書き改められる。其処ではPSは今の自己からPS3→PS4の順に成立する。其して、PS2やPS2-1はそれぞれ成立する。この仕組みは、第5章で述べる要件を定義する公式(PRD)では、PS4(PS2-1)→PS2→PS3の順に成立する。PS4(PS2-1)は、PS2-1がPS4と同じ様に扱われる事を意味する。

TDMが脈絡と同等の役割を担う為には、脈絡が例えばわれわれの思考作用が果敢なく継続されている事を想起すれば、TDMも同じ様に果敢なく作用をし続けなければならない。其れは、上述の個別座標軸の回転、並びに全座標軸の回転の事である。其の作用は以下の様に言い換える事が出来る。

(01)PS3の集合化と其の網羅

(02)PS2の集合化と其の網羅

(03)PS4の集合化と其の網羅

(04)PS3, PS2, PS4の集合の網羅

[本発明の内容]

#### 第4章 仮想空間の実装構造(SF)

前章のTDMは、意識を表す存在を捉えるモデルになっている。其れは、思考の限界を克服するモデルの仕組になっている。

[0206] 本章ではTDMの仕組の意義を電算機を用いて実証する為に、其の仕組を電算機に適用可能となる様に翻訳し直す。翻訳し直されるTDMの仕組をSFと記す。

[0207] 本章では、TDMをSFに翻訳する必要な事柄が述べられる。

以下に述べる事柄に関しては、部分的に非特許文献1〜9で述べられている。

##### 4.1 要件

意識を表す存在は、要件の根源として位置付ける事が出来る。意識を表す存在から解き放される存在は、要件を構成する存在として位置付ける事が出来る。

[0208] ソフトの場合、要件を構成する存在は主語として位置付ける事が出来る。主語は、名詞によって代表される。例えば、本研究で言う存在の実体が名詞である。其して、其の属性は名詞に帯同する存在である。主語とは、この関係によって定義される存在の事である。即ち、要件={主語}である。

ソフトの場合、其のシステムに拘わる例えば画面、帳票、ファイル、電文等は、主語の集合として位置付けられる。即ち、其れらは要件を形成するものである。

#### 4.2 主語

主語は、名詞、狭義の属性、並びに広義の属性で定義される。

[0209] 即ち、主語=[名詞、狭義の属性、広義の属性]である。

##### (01)名詞の種別

本研究では、要件に属す名詞は名称と其の識別子、並びに名詞種別で定義される。

[0210] 即ち、名詞=[名称、識別子、名詞種別]である。

[0211] 本研究では、名詞の種別は22種となる。其の定義は表2に示される。名詞の種別が22種になる理由は、PSの構造とソフトを受け入れる電算機の仕組の関係から決定される。

##### (02)名詞の狭義の属性

名詞の狭義の属性は、主語を種別化する要因になる。

[0212] 名詞の狭義の属性は5種である。其の定義は表3に示される。名詞の狭義の属性が5種になる理由は、PSの構造とソフトを受け入れる電算機の仕組の関係から決定される。

##### (03)名詞の広義の属性

名詞の広義の属性は、主語を定義する要因になる。

[0213] 名詞の広義の属性は10種である。其の定義は表7に示される。名詞の広義の属性が10種になる理由は、PSの構造とソフトを受け入れる電算機の仕組の関係から決定される。

##### (04)主語種別



主語は、要件を捉える最小単位になる。其の種別は30種である。

[0214] これは、ベクトルの種別に対応するものである。其の定義は表1に示されている。

[0215] 主語種別が30種になる理由は、PSの構造、ソフトを受け入れる電算機の仕組、並びに要件の性質の関係から決定される。しかし、其の主語の種別の定義は、要件の影響を受ける事はない。

#### 4.3 ベクトルの原型

前章で定義されたPSは、SFではベクトルと呼び換えられる。其の場合のベクトルの標準形は図22に示される。これは、ソフトのプログラムの性質と其れを受け入れる電算機の仕組の関係から求められる定義で、更に14種に翻訳し直される。表4参照。其の用途については、以下に述べる。

##### (01)L4(論理要素)

未来の自己を表す名詞に真偽値を決定する為の叙述である。

[0216] PS4から定義されるものである。

##### (02)L2(論理要素)

子孫を有しない自己を表す名詞に真偽値を定義する為の叙述である。

[0217] PS2から定義されるものである。

##### (03)L3(論理要素)

過去の自己を表す名詞に真偽値を決定する為の叙述である。

[0218] PS3から定義されるものである。

##### (04)I2(入力作用要素)

上記L2の成立を支援する為の叙述である。

##### (05)O4(出力作用要素)

上記L4の結果を宣言する為の叙述である。

##### (06)S4(同期作用要素)

ベクトルの領域を用いてSFの周期性を整備する為の叙述である。

##### (07)R4(経路作用要素)

W04とW02に順序性を与える為の叙述である。W04とW02については、本章4.

8で述べる。

(08)R2C(経路作用要素)

W02と隣下位のW04に順序性を与える為の叙述である。

(09)R2(経路作用要素)

W02とW03に順序性を与える為の叙述である。W03については、本章4. 8で述べる。

(10)R3R(経路作用要素)

W03とW04に順序性を与える為の叙述である。

(11)R3E(経路作用要素)

SFの作用を終了させる為の叙述である。

(12)R3C(経路作用要素)

W03と隣下位のW04に順序性を与える為の叙述である。

(13)R3D(経路作用要素)

W03と隣上位のW03に順序性を与える為の叙述である。

(14)R3M(経路作用要素)

W03と隣々上位のW04に順序性を与える為の叙述である。

#### 4. 4 ベクトルの集合

14種のベクトルは、TDMの座標の役割を基にすれば、PSの構造、ソフトを受け入れる電算機の仕組、並びに要件の性質の関係から、以下の様に集合化される。これは、TDMの擬似空間に対応するSFの擬似空間を定義する座標の役割を果たす。

(01){L4}, {04}, {S4}, R4}

(02){L2}, {I2}, R2C, R2}

(03){L3}, R3R, R3E, R3C, R3D, R3M}

SFでは、上記(01)をW04、上記(02)をW02、上記(03)をW03と記し、パレットと総称する。

#### 4. 5 ベクトルの定義規則

主語に対応して、ベクトルは定義される。其の定義の仕方は普遍的な規則になる。  
以下に、ベクトルが定義される手順を示す。

(01)要件から名詞を探す。

(02)其の名詞の種別を表2(名詞の種別)を基に決める。

(03)其の名詞の狭義の属性を表3(名詞の狭義の属性)を基に要件から探す。

(04)上記(03)の結果、主語の種別を表1(名詞種別と主語種別(ベクトル種別)の関係)を基に決める。ベクトルの定義規則は、73種類(表1)となる。其の理由は、PSの構造、ソフトを受け入れる電算機の仕組、並びに要件の性質の関係から決定される。

(05)其の名詞の広義の属性を表7(名詞の広義の属性)を基に要件から探す。

[0219] 名詞の広義の属性は表7で定義されているので、主語の種別が求められれば其の名詞の属性を要件から探す事が出来る。本研究で言うベクトルの定義規則とは、この事を指している。ベクトルの定義規則は、本論文の資料1として添付する。

名詞の属性が求められれば、ベクトルは決定する事が出来る。其の構造はPSと同じである。其の標準形を図22に示す。14種のベクトルの原型は、これを基に定義される。

従来では、要件を機能的に把握し、其れがプログラムに置き換えられている。要件を機能的に把握する為には、要件の成立の由来を認識する事が求められる。其れは、求めるべきゴールが段階的な作業を得なければ、到達し得ないと言う事を意味する。この事が要件を定義する作業、プログラムを作る作業、保守をする作業、テスト作業等が非能率になる原因になっている。

其れに対し、本項で示している様にベクトルを定義する作業に注目すれば、求めるべきゴールに必要な情報は既に既知なる規則を用いて要件から抽出する事が出来る事である。もし、抽出されるべき情報が其の要件の中に見当たらなければ、其の情報は不要であるか、または欠落しているかのいずれかである。この様な対応が、要件に関する成立の由来の認識とは別に可能となる事である。

#### 4. 6 パレット関数

第3章3. 6で述べた様に、TDMが脈絡と同等の役割を担うためには、次の役割が必要である。

(01)PS3の集合化と其の網羅

(02)PS2の集合化と其の網羅

(03)PS4の集合化と其の網羅

(04)PS3, PS2, PS4の集合全体の網羅

上記(01)のPS3の集合化とは本章4. 4(03)の事で、其れは座標軸と見立てられるW03の事である。上記(02)のPS2の集合化とは本章4. 4(02)の事で、其れは座標軸と見立てられるW02の事である。上記(03)のPS4の集合化とは本章4. 4(01)の事で、其れは座標軸と見立てられるW04の事である。

[0220] 其れらを網羅する作用をパレット関数と記す。其れらは $\Phi 3$ ,  $\Phi 2$ ,  $\Phi 4$ で表される。其の標準形を図26に示す。

#### 4. 7 パレット連鎖関数

第3章3. 6で述べた様に、TDMが脈絡と同等の役割を担うためには、上記(04)の役割が必要になる。其の作用をパレット連鎖関数と記す。其れは $\Phi 0$ で表される。其の標準形を図25に示す。

[0221] パレット連鎖関数の役割を能率化させる為に、経路制御情報テーブルと記す定型的なテーブルが用いられる。其の標準形を参考の為に表8に示す。このテーブルは、PRD情報が定義されれば機械的に定義される。

#### 4. 8 パレット

本章4. 4で集合化されるベクトルの集合とパレット関数をパレットと記す。以下に其の定義を示す。

$$(01)W04i = \Phi 4[\{\{L4\} + \{04\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

添え字のiは、パレット、並びにベクトルの識別を示す。記号+は、パレット関数で管理されるベクトルの実行順序を示す。

#### 4. 9 SF

SFは、TDMが電算機で可用させる事が出来るように再定義されたものである。以下の様に定義される。

[0222]  $SF_i = \Phi O[W04 + W02 + W03]_i$

添え字の*i*は、SFの識別を示す。記号+は、パレット連鎖関数で管理されるパレットの実行順序を示す。

結果的に、SFは意識を表す存在を電算機上で擬似的に成立させる構造となる。其の元になる情報は、要件である。

W04, W02, 其してW03は、TDMの座標の概念に相当する。

[0223] 本研究ではこれらをパレットと総称する。ベクトルは、其の役割に基づいて、これらパレットの要素に割り当てられる。パレットとパレットを結ぶ作用は、経路作用要素と呼ぶベクトルの役割で成立する。SFの概念図を図23に示す。

プログラムの構造については、ウィナー、ダイクストラの研究がある。本研究で求められるSFは、プログラムの構造と言う意味では全く同じ概念であるが、其の由来は前者のものとは異なっている(非特許文献10及び11参照)。

## 第5章 要件を捉える公式(PRD)

SFをソフトを開発する際の思考法にする為に、パレットに属す作用要素と呼ぶベクトルが定義される。其の事により、SFはソフトの要件を表す本研究で言うところの1個の意識を捉える仕組みとなる。其して、SFは其の構成パレットに属すベクトル、パレット関数、並びにパレット連鎖関数をプログラム言語で定義する事により、其の要件をプログラムとして表す事が出来る。

従来では、要件からプログラムが導き出されるのに対し、既に述べた様にSFは要件から独立している本研究で求められるベクトルを定義する規則を用いて要件を探索し、其の事によりプログラムを決定する事が出来る。換言すれば、SFは要件を捉える思考作用と等価に位置付けられる。

本研究では、SFを要件を捉える要素と見立て、更に拡張される要件を捉える方法に翻訳し直される。翻訳し直されるSFをPRDと記す。本章では、SFをPRDに翻訳する必要な事柄が述べられる。

### 5.1 PRDの構造

3個のSFで成立するPRDの参考例を図24に示す。1個のSFにおける3種のパレットを結ぶ経路作用要素、R4, R2, R3Rは、例外なく必然的に定義される。其して、3



個のSFを結ぶ経路作用要素, R2C, R3C, R3D, R3Mも例外なく同様に必然的に定義される。

- [0224] 必然的に定義されるという意味は, 要件を捉えるL4, L2, L3のベクトルが定義されれば, 其れから機械的に誘導されて定義されるという事である。

1個のSFが1個の意識を表す存在を捉える構造である事を留意すれば, この例は, 3個の意識を表す存在を捉える構造を含意している。付言すれば, この例は3個の意識で成立するソフトである。図で示す入力定義体, 出力定義体, 並びに其れらの関係はこの3個の意識に端を発する脈絡の上に現れる存在を意味する。更に, これら入力定義体, 出力定義体に属す名詞も, 其の脈絡の上に現れる存在を意味する。

- [0225] 入力定義体, 出力定義体, 並びにそれらに属す名詞の関係は, 其れら名詞の属性により定義されるベクトルの相補作用によって成立する。

要件により発生するSFの数, 並びに其れらを結び成立するPRDの形態は, 発生するSFの役割, 並びに性質により様々な形態となる。しかし, SFを結ぶ経路作用要素は, 其の事とは関係なく普遍的に定義される。

意識を表す複数の存在を内包する唯1つの意識を表す存在が成立する場合もあり得る。其れが, 複数のSFで構成されるPRDが唯1個のSFに集約され得る事をも意味する。反対に, 1個のSFが複数のSFに展開される場合も成立するのは当然の帰結である。

この構造は, 従来のプログラムの構造と異なっている。其の考えは, 例えば非特許文献12等を参考にすればより明らかである。

## 5. 2 SFの結合規則

複数のSFを結合する規則を以下に示す。この規則は, 複数の意識の成立順位を規約する作用に相当する。

## (01)SFの継続規則

隣り合うSFを順序付ける結合は、R2C、R3Cで行われる。即ち、R2CはW02から隣り合う下位のSFのW04を結合し、R3CはW03から隣り合う下位のSFのW04を結合する。

## (02)SFの多重規則

隣り合うSFを逆に順序付ける結合は、R3Dで行われる。即ち、R3DはW03から隣り合う上位のSFのW03を結合する。

- [0226] 脈絡で成立する存在には、其の全てが記憶を成立させる存在にはなり得ない事態が生じてしまう。即ち、其の一部が同化を成立させる存在になってしまうからである。其の為に言表が不充分になり、其れによる曖昧さが生じる。自己なる存在は、其れを解消する為に本能的に其の為の意識を表す存在を成立させる。この役割がR3Dで表される。

## (03)SFの重複規則

直列に並ぶ3個のSFの隣り合うSFを逆に順序付ける結合は、R3Mで行われる。即ち、R3MはW03から隣々上位のSFのW04を結合する。上記(02)の事態が重複して生じる場合、自己なる存在が果たす役割がR3Mで表される。

## 5. 3 端点と始点の関係

ベクトルの第2規約の属性は、自己の名詞を基に、自己、或いは其れ以外の名詞を用いて定義される。この場合の自己の名詞を端点、其れ以外の名詞を始点と記す。

- [0227] 始点となる名詞の利用の仕方について、SF内、並びにSF間で以下の関係が成立する。この関係は、本研究の仮説の基で成立する。本論文では、其の論述を行わない。

(01)L4の始点となる名詞は、同じSFのL4の名詞を用いる事が出来る。

(02)L4の始点となる名詞は、同じSFのL2の名詞を用いる事が出来る。

(03)隣り合うSFで上位に位置するSFのL4の始点となる名詞は、下位に位置するL4の名詞を用いる事が出来る。

(04)隣り合うSFで上位に位置するSFのL4の始点となる名詞は、下位に位置するL2

の名詞を用いる事が出来ない。

(05) 同じ位置に配置される複数のSFのL4の始点となる名詞は、同じ位置に配置されるSFのL4の名詞を用いる事が出来ない。

(06) 同じ位置に配置される複数のSFのL4の始点となる名詞は、同じ位置に配置されるSFのL2の名詞を用いる事が出来ない。

(07) 直列に並ぶ3個のSFにおいて、其の最上位に位置するL4の始点となる名詞は、其の最下位に位置するL4の名詞を用いる事が出来ない。

(08) 直列に並ぶ3個のSFにおいて、其の最上位に位置するL4の始点となる名詞は、其の最下位に位置するL2の名詞を用いる事が出来ない。

(09) 上記(07)のL4の始点となる名詞が、最下位のSFのL4の名詞を用いる場合には、其のSFのL4の名詞は同じ名詞を最上位のL4の名詞として別個に配置する。

其の最下位のSFの其のL4は、別個に配置される名詞に対して、同じ作用を行う。

(10) 上記(07)のL4の始点となる名詞が、隣々下位のSFのL2の名詞を用いる事が出来ない。

(11) 隣り合うSFで下位に位置するSFのL4の始点となる名詞は、上位に位置するL4の名詞を用いる事が出来ない。

(12) 隣り合うSFで下位に位置するSFのL4の始点となる名詞は、上位に位置するL2の名詞を用いる事が出来ない。

(13) L3の始点となる名詞は、同じSFのL2の名詞、L3の名詞、並びに既に過去である事が明らかなL4の名詞を用いる事が出来る。

(14) L3の始点となる名詞は、同じSFの未来となるL4の名詞を用いる事が出来ない。

(15) L2の始点となる名詞は、同じSFのL2の名詞以外は存在しない。

#### 5. 4 入力定義体と出力定義体の相補関係

SFを定義する主要因は、入出力定義体である。これを成立させる作用が入出力作用要素のI2, O4である。入出力作用要素は、TDMと電算機システムの概念を融合する観点から定義されている。

O4の役割は、意識を表す存在から脈絡を成立させる存在を解き放す作用に相当

する。

[0228] I2の役割は、意識を表す存在を成立させる起因なる存在、即ち、自己なる存在(図11)を成立させる作用に相当する。

この観点では、入力作用と出力作用には、役割の上から大きな隔たりのある事が分かる。しかし、I2は自己なる存在(入力定義体)を成立させ、O4は其れを基に新たな存在(出力定義体)を成立させる事において、両作用には密接な繋がりがある。この関係を入力定義体と出力定義体の相補関係と記す。

SFでは、W02に入力定義体が属し、W04に出力定義体が属している。其して、SFはこれら定義体により定義される関係になる。この関係は、SF其れ自体も自己なる存在と等価である事を裏付けている。其して、入力定義体と出力定義体が相補関係を成すとは、其のSFが果たす自己なる存在の思考作用が其のSFで閉じていなければならない為の条件になる事である。

換言すれば、入力定義体と出力定義体の間にこの相補関係が成立する場合に限り、SFが成立する事を意味する。即ち、複数のSFが必要となり、且つ、其れらが結合されるのも、其れらに係る全ての定義体に相補関係を成立させる為である。

#### 5. 5 SFの縮約規則

PRDに属す複数のSFの間で、其れらが統合される関係を縮約と呼ぶ。以下に其の可否条件を示す。以下の関係と本研究の仮説の関係は、本論文では省略する。しかし、其れらは仮説の基で成立する関係である。

(01)2つのSFに同じ入力定義体が定義される場合、其れらSFは縮約される。何とならば、其の定義体により成立するSFは、お互いに同じ自己を定義している事になるからである。

(02)2つのSFに同じ出力定義体が定義され、且つ、入力定義体が異なる場合、其れらSFは縮約する事が出来ない。何とならば、其の定義体により成立するSFは、お互いに異なる自己を定義している事になるからである。

(03)2つのSFで出力定義体が異なっても、上位の入力定義体に対し、下位の入力定義体が相補する関係を成すならば、其れらSFは縮約される。この場合、下位のSFの出力定義体は上位のSFの出力定義体として配置される。

(04)上記の関係は、並列に並ぶSF間においても成立する。

#### 5.6 ベクトルの座標

TDMは、意識を成立させる存在を捉える為の擬似的な3次元空間の役割を果たしている。SFもこの関係は同じである。其れ故、TDMの座標軸に相当するパレットは、其のパレットに属すベクトルに座標を与える役割を果たしている。其して、SFは其れを構成するパレットにより擬似空間を定義する関係になっている。

[0229] この擬似空間に浮かぶ存在は名詞であり、脈絡で現れる存在に相当する。SFが作用する事により其れら名詞に脈絡が生じ、其の状態がわれわれが認識するロジックに相当する。其のロジックが意識を表す存在を捉えると言う構想である。複数のSFが存在するPRDは、この擬似空間が複数成立する関係を表す。換言すれば、要件はこの擬似空間によって捉えられると言う関係である。

PRDでは、この座標はPRD識別子、SF識別子、パレット識別子、ベクトル識別子の合成情報として定義される。この事により、ベクトルを定義する端点、始点の関係が要件の論理性を深く理解しなくとも、機械的に捉えられる事になる。

#### 5.7 テンプレート(TMP)の意義

ベクトルをプログラム言語で機械的に定義する仕組のプログラムをテンプレートと記す。テンプレートは、プログラム言語毎に定義される。本論文では、資料2としてこれを添付する。テンプレートの意味する所は、上述のベクトルを機械的に定義する仕組の他に、其れがPRDを実際的に成立させる証明と成る事である。即ち、ベクトルの定義規則は、PSを基に定義される。

[0230] PSは、脈絡のモデル(思考モデル)から導かれている。脈絡のモデルは、自己なる存在のモデルから導かれている。自己なる存在のモデルは、本研究の仮想存在から導かれている。

[0231] 以上の事から、このテンプレートによりPRDが定義可能となる事は、仮想存在に用いられる仮説に妥当性がある事を意味する。

#### 5.8 適応化の幾つかの工夫

本方法を具体的課題に適用する上で、以下の様な工夫が有効となる。

(01)経路作用要素の定義を簡明にする為に、パレット連鎖関数とベクトルに分ける。



其れに付随して、経路制御情報テーブルを設ける。表8参照。

(02)経路作用要素の定義を簡明にする為に、R3C, R3D, R3Mをパレット連鎖関数に担わせる。

(03)同期作用要素の定義を簡明にする為に、其の一部をパレット連鎖関数に担わせる。

## 5. 9 本方法論の総括

本研究で求められるベクトルとは、要件に属す名詞を実体とする主語の事である。主語は、名詞と其れに帯同する属性として定義される。其して、其の構造は7種の叙述文と4種の領域で定義される。其の領域によりベクトルは名詞の存在を言表可能とし、且つ、其のベクトル自体を他のベクトルから独立させる事が出来る。以上の事により、ベクトルは要件を捉える最小クラスとして位置付けられる。この最小クラス概念が要件、ソフト、並びにプログラム上で成立する事により、ソフトを捉える概念が従来に対し改善される。其の事を以下に述べる。

### (01)従来のアプローチ

図28は、従来のアプローチを概念的に示した図である。

[0232] 従来の方法論、例えば、DOA, FOA(機能指向法), OOAの根底に置かれる思想は、要件を定義しつつ其れを何らかの基準、例えば知識を基に分割し、実行環境に適応出来る様に補正、即ち設計して、其れをプログラム化する方法である。この方法では、結果的に、要件を分割する行為が分割者の恣意によって行われてしまう。

[0233] 其の事により、分割される事柄の役割、性質が、他の分割される事柄の役割、性質と完全に分離する事が困難になる。

[0234] 其の恣意は、本能的に其の分割者の論理的思想を基に成立するものである。論理的思想は事柄を紡ぐ作用となり、其れを分断する作用とは相反する傾向を持つ。其の事が要件を分割する事態を余計困難にさせている。即ち、要件の分割は均一なクラス概念を成立させる事が出来ないと言う事態を生じさせる結果になっている。

[0235] ソフト開発における諸問題は、この事態を克服するところに其の病巣がある。本研究の最小クラス概念は、この本質的な問題の解決に寄与しうる事は明らかである。

[0236] 図20では、従来の要件の捉え方と本方法論の要件の捉え方の相違が示されてい

る。

(02)本方法で定義されるプログラムの位置付け

本方法で定義されるプログラム(SF, またはPRD)は, SFの定義から其れを実装する工学的環境, 即ち, 入出力作用要素を除き, 定義される。其れは, 従来の方法によりプログラムが成立する位置と異なるものである。其れを図21に示す。

[0237] これは, 本方法論によるプログラムの役割が従来のプログラムの役割と必ずしも同じでない事を意味する。本方法論によるプログラムは, 要件を捉えるプログラムとして位置付けられる。従来のプログラムは, 要件をプログラムで再定義する役割として位置付けられる。即ち, 従来のプログラムは要件に矛盾がない事を前提に定義される事になる。其の為に, 事前に要件の妥当性を検証しておく必要がある。

[0238] 其れに対し, 本方法論によるプログラムは, 従来のプログラムと同じ様に電算機上で可動可能であるが, 要件の矛盾を発見する仕組みになっている。付言すれば, 本方法論のプログラムのソースライン数は従来のプログラムのソースライン数に較べ概算7倍〜10倍となるが, 其の役割の中には従来のプログラムが担っていない要件の矛盾を発見する仕組みも含まれているのである。

[0239] しかし, 表9で示されている本方法論による開発実績は, 本方法論のプログラムのままで実用に寄与している。因に, 経過時間性能では殆ど問題は生じていない。しかし, CPU占有率は増加する。仮に, ソースライン数の多さが気になるのであれば, 其れは論理的に再定義し直せば済む問題である。

本方法論のプログラムは, 要件を迅速に把握し, 其の妥当性を電算機上で迅速に確認する事が出来る。即ち, 図22で示す様に, ベクトルの第6規約では主語の不在が開示される。この状態が要件に潜む矛盾を指摘する事になる。其の意味で本方法のプログラムは, 要件を定義する文章として位置付けるのが妥当である。

[0240] 本プログラムが果たすこれらの効果は, 最小クラス概念から誘導されるものである。

(03)図22で示すベクトルの第1規約では, 主語の第4領域の値の有無が確認される。其れは, 名詞の真偽値を判定する為の叙述である。真値が既に決定していれば, ベ

クトルは繰り返し実行される事がない。この配慮は、必然的に電算機の処理時間を短縮させる役割を担う。この第1規約は、要件が名詞の集合である事に留意すれば、其の叙述は要件の妥当性を捉える役割を果たすものである。

[0241] しかし、この叙述は要件の論理性とは無関係に定義可能となる。

[0242] この効果は、最小クラス概念から誘導されるものである。

(04)図22で示すベクトルの第3規約では、主語の第2領域の値の有無が確認される。其れは、名詞の真値の成立を判定する為の叙述である。この叙述は、ベクトルを成立させる名詞の属性を検証している事になる。

[0243] しかし、この叙述は要件の論理性とは無関係に定義可能となる。

[0244] この効果は、最小クラス概念から誘導されるものである。

#### (05)要件を捉える公式

本方法論では、要件は第4章4. 1で述べられる様に主語の集合となる。其の形式で要件が定義されれば、上図の通り、本方法論のプログラムは機械的に決定される。上図の構図が本方法論で言う要件を捉える公式を意味する。

#### (06)本方法のアプローチ

本方法の思想は、従来方法の思想よりもより根源的な水準でソフトを捉えている事は明らかである。其の事において、本方法の思想はこれまでの方法を統一する事になる。

[0245] 従来の方法が妥当性、並びに検証性を中心とする開発方法となるのに対し、本方法では其の問題は上述(03)(04)で述べている様にベクトルに置き換えられる。

### 第6章 結論

本研究の意義を予知する為に、これまでのソフト世界の現象を考察する。

(01)ソフトを体系化する原理が見いだされていない。其の結果、以下のような現象が散見される。

01. 自己で作成したプログラムの正当性を自ら確信する事が難しい。

02. 技術的な問題が管理の問題に置き換えられて議論される傾向が強い。

03. 関係者は次第に当事者能力を失う傾向がある。

04. 技術的問題が世間的常識で律される傾向がある。

05. ソフトでは、妥当性、並びに検証性の保証が総合的に求められる。

06. 開発の生産性は能力の低い属人性に引き下げられ易い。

07. ソフトの維持費用は効果と関係なく経時的に高騰する。

08. プログラムの市場価値を分かりにくくしている。

09. プログラムの生産コストの尺度が定まりにくい。

(02)上記現象の直近の原因を求めると、以下の2点に纏められる。

01. 要件、並びに論理は、合成されると曖昧さ(合成の誤謬)が生じる。

02. 要件、並びに論理は、恣意的(闇箱)になる。

(03)ソフト開発では、どの工程の作業でも、結果に現れない作業がより高い割合を占めている。

(04)以上の事から、ソフトは強く意識に根差して成立するものである様に思われる。単に、工学的手法だけで克服出来る課題ではない様に思われる。

以上の観点から、ソフトの世界を合理的に構築する為には、結果に潜む事態を解明し、其れに対処出来る施策を講じることが必要である。

[0246] 譬え、説得力のある論述であっても、其れが工学的観点に立つ限り、ソフト世界の課題を克服する手段となる事は有り得ない。これまでのソフトの世界は、ハードウェアを中心とする工学的世界に依存して来た為に、工学的世界から独立し得なかった様に思われる。

[0247] また、他の分野の知識にも依存してきた事が、ソフト世界の独立を妨げてきた原因である様に思われる。

例えば、工学が公共的認識を作り出す方法を提示する世界であるとするなら、ソフトは、例えば意図、意識などの語彙で代表される朦朧とした世界を捉える方法を必要とする世界である。例えば、科学的な在り方とは視点の異なる新しい認識世界の科学と位置付ける事が出来る。

[0248] 今後のソフト世界に望まれることは、純粹にソフト世界を捉える思考方法を発見する事である。

しかし、この提案は関係者にとり、受け入れ難いであろう事は察せられる。何故なら

、既存のソフトの世界は、何処かで破綻しない限り改善出来ない所まで社会構造的な仕組みが作り上げられ、殆どの関係者は其の中に組み込まれているからである。

[0249] 其の事を実証する世界規模の潮流を以下に示す。

(01)世界的なY2K騒動は、関係者に多くの教訓を与えた筈である。

[0250] しかし、其の後、何かが変革されたであろうか。既に、あの騒動は忘れ去られている様に思われる。其して、関係者の責任感、自覚は霧散してしまっている。

(02)コンピュータ利用の殆どの機関は、世間的な問題を起こしても免罪符が与えられる機関と協調している。

(03)コンピュータ利用の殆どの機関は、自己のソフト部門を切り離す傾向にある。ソフトの本質が理解されれば、信じ難い振る舞いである。

(04)精神性を伴うわれわれの思考行為を、例えば、人工知能、ロボット等の機能的概念と同一視する風潮が作り出されている。

本研究は、ソフトの本質がわれわれの存在と深く関わる関係にあるとして捉えられる事を示している。其して、本研究は、人が存在しない世界でソフトは成立しない。また、存在する人が気付かない世界でソフトが成立している事を説明しようとしている。

本研究の成果は、この思想に基づいて論考された結果である。例えば、要件の曖昧さを回避して要件を定義するベクトルは、われわれの思考法を成立させる最小単位の発見に匹敵する概念である。この事によって、以下の問題を克服する原動力に成ると思われる。

(01)われわれの思考法に生じる曖昧さの問題

(02)論理の中に潜む恣意的な問題

(03)言表不可能な事態に対処する問題

其れは、本研究の成果を用いて既に開発されているシステムを考察すれば、其の事を予感する事が出来る。正常な評価は、現在の状況では成し難いと思われるが、本研究の成果である概念を認識すれば、其の効果も容易に認識する事が出来る筈である。

本研究の成果であるPRDを考察すれば、われわれの論理的史観の深層に位置する性質を捉えている事が分かる。この性質は、論理的史観を前提とする立場で捉える



事は困難であるかもしれない。しかし、既存のプログラムから普遍的なアルゴリズムでPRDを捉える事が出来る。この場合のこのアルゴリズムは、本研究の成果から得られるものである。即ち、本研究の成果は、既存のソフトの世界とは無縁ではなく、深層において繋がっている。この事は、本研究の成果が既存のソフトの世界の課題を克服する為に用いる事が出来る事を示している。

以上の本研究の成果は、ベクトルの定義規則が種別化された主語(要件に属す名詞)により決まり、其の定義規則に従えば、最小クラスの機能が求められる。其して、主語の種別数、ベクトルの定義規則の数、其れによって決まる最小クラスの機能の種別は、全て既定値となる事によるものである。

付言すれば、要件を捉える為のわれわれの思考法に改善を齎す。例えば、要件に関する関係者の認識から曖昧さを除く事が出来るので、能率的に要件に関する合意を作り出すことが出来る。ソフトの世界に最小クラス概念を導入する事は、要件を捉える為のわれわれの思考の単位となり得るだけでなく、プログラムを定義する1命令文までを公共化する効果を獲得する事が出来る。

ソフトの世界は、半世紀に及ぶ経験を有している。しかし、本章の冒頭で触れた様に、其れを捉える原理の探索は放置されたままである。例えば、飛行機は50年を得て音速を突破した。無線技術は、30年で完成したと言われている。近代におけるこのような工学的世界の変化には、其れなりの状況があり、其れを受け入れる人的土壌が其の分野にあった筈である。其の事を思うと、この分野の原理的な認識は存在していないので、混沌としたまま継続されている。

本研究を通じて明らかになった事は、ソフトの世界には原理、公理、理論が育つ土壌が見出し難い事である。換言すれば、基礎的な科学は歴史を振り返れば、個人に端を発している。しかし、この分野ではこの原則を作り出す事が非常に難しい様に思われる。しかし、この手順を踏まなければ、ソフトウェアは科学と呼ばれる期待される世界に到達する事は難しいであろう。

この分野の全ての関係者に求められる事は、科学の歴史に学ぶ事である様に思われる。換言すれば、今の時代、ソフトの専門家ひとりひとりが賢明な判断を行い得る決意を養う事が重要であると思われる。

本研究の成果は今後様々な分野で利用されると思われるが、其れに際して様々な視点からの議論が期待出来る。其れも、本研究の齎す成果だと考えられる。

本稿の最後に、本研究の全体像を要約する。

[0251] 本研究の目的は、思考の限界を克服する思考モデルを形而上学的な仮説を用いて定義する事である。其して、其の妥当性を実感する事である。其の実感を得る手段として電算機が用いられる。

[0252] この為に、この思考モデルは電算機上で定義される。

[0253] 換言すれば、この思考モデルは電算機上で定義可能となるように定義される。

結果から言えば、この思考モデルを定義する仕方は電算機のソフトを開発する場合の思考法を反映する。其して、本論で述べている様に、其の定義の仕方には公式と見なせる普遍性が成立する。

[0254] これは本研究の成果である。

[0255] この普遍性の証明は、結果的に添付されているベクトルの定義規則(資料1), TM P(資料2)で与えられる。

[0256] 其処には例外のない規則が論述されているだけである。

本研究では、思考は自己に成立する意識の作用として定義される。其して、この作用をモデル化する為に本研究では存在論をモデル化する。其れは存在, 意図, 自己, 意識をモデル化する事である。

本研究では、思考は存在論の部分を成す作用であると考え。其の為、存在論の全体をモデル化する必要がある。そうでなければ、論理的に妥当な思考モデルを定義する事が出来ないからである。本研究が長期にわたる時間を要した理由はこの為である。

本研究の存在に関する思想性は基本概念(第2章2. 3), 存在の概念(第2章2. 4. 1)で明示されている。其して、本研究では其れらを裏付ける為に集合論の概念を用いて、モデル化が行われる。

本研究では、存在は連鎖として定義される。其して、連鎖の要素は論理原子である。論理原子は概念であり、その概念は本研究で仮説されている。其して、論理原子は本研究で仮説される時間速度から導かれる。

本研究では、存在は集合の概念を用いてモデル化される。この為に、時間速度、並びに論理原子は重要な位置を占める。

- [0257] 時間速度は、限定的で無次元の有理数である。其して、本研究では存在が空間を占める概念であると考え事から、論理原子は空間を表す有理数として位置付けられる。

本研究では、意識は自己に内在する連鎖である。

- [0258] 其して、本研究は、思考を以下の様に定義する。即ち、意識に内在する連鎖が自己を介して開放される様相が思考の作用である。

本研究では、思考の作用を脈絡と定義する。脈絡に属す全ての連鎖が言表出来れば、其の全言表の集合は其の思考の限界を越える作用を成立させる事になる。これが、本研究で行われる存在論に関する論考の結論である。

本研究では、叙述と言表の用法を区別している。叙述は存在の定義を意味し、言表は存在が存在する事の証明を意味する。叙述は時間的に過去の存在を定義する事が出来る。

- [0259] しかし、言表は時間的に未来の存在しか証明する事が出来ない。

脈絡に属す連鎖の成立を証明する為には、其の連鎖の過去と未来の連鎖の言表が成立する事が必要である。何とならば、自分で自分を証明する事は出来ないからである。其れ故、今の連鎖の過去の連鎖が未来の連鎖と同じ位置付けになる様に、脈絡の構造は修正される。其れがENWである。其して、言表の構造がPSである。ENWの全体を言表する構造がTDMである。思考モデルの仮説を電算機を用いて実証する基本構造がSFである。其して、其れを一般的に実証する構造がPRDである。ENWの全体を言表する構造は、TDMから与えられる二つの制御プログラム( $\Phi 0$ ,  $\Phi P$ )である。これらプログラムには、普遍性が成立する。

PSはSF, PRDにおいては要件を捉える最小単位に変わる。其れがベクトルである。ベクトルの定義の仕方は普遍的な規則になる。其れをプログラム言語で言表したものがTMPである。

- [0260] 即ち、電算機上で実証される思考モデルは二つの普遍的な制御プログラム、ベクトルで定義する事が出来る。其して、其の思考モデルは要件を満たす役割を電算機上

で果たす。

[0261] 其の働きが妥当であるならば、本研究における仮説もまた、妥当性であると考えられる。

資料1, 資料2は、其の妥当性を証明する役割を果たす為のものである。

なお、本発明は、上述した実施形態および実施例には限定されず、本発明の技術思想の範囲内で様々な変形が可能である。たとえば、ビジネス・メソッド、ソフトウェア開発装置、ソフトウェア開発支援装置、ソフトウェア開発管理装置、あるいはこれらの機能をコンピュータに実現するためのソフトウェア並びに該ソフトウェアを搭載した記録媒体・専用機、などとしてもそれぞれ実現することが可能である。さらに、上記で説明したように本発明は、方法としても、或いはかかる機能を備えるソフトウェアとしても、該ソフトウェアが搭載される装置・ツール(ソフトウェア自体の場合も含む)としても、さらにはシステムとしても、実現され得るのはもとよりである。

[0262] 例えば図212は、本発明の異なる一実施形態として、「開発対象のソフトウェア」を生産するためのプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置のいずれかとして本発明を実施する場合に機能として備える構成を示した機能ブロック図である。同図に示すように、本発明に係るプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置は、全体制御部2010、PRD(プロセス・ルート・ダイアグラム)部2030、ベクトル形成部2040、パレット部2050、パレット連鎖関数部2060及び情報記憶部2070を備えて構成される。

[0263] 全体制御部2010は、本プログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置の全体の動作制御、タイミング制御、入出力制御等を行う機能を有しており、当該機能を持つ専用チップ、専用回路、または当該機能をコンピュータに果たさせるためのソフトウェア

(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現される。

[0264] PRD(プロセス・ルート・ダイアグラム)部2030は、名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉えるための機能を有しており、当該機能を持つ専用チップ、専用回路、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現される。

[0265] ベクトル形成部2040は、定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム):

L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E

を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで該ベクトルをサブルーチン・プログラムとして形成するするための機能を有しており、当該機能を持つ専用チップ、専用回路、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現される。

[0266] パレット部2050は、サブルーチン・プログラムとして形成したベクトルを、基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

に集合させるための機能を有しており、当該機能を持つ専用チップ、専用回路、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現される。

[0267] パレット連鎖関数部2060は、基本構造ごとに集合されたパレットの実行を管理する



パレット連鎖関数を用いてパレットをプログラム化するための機能を有しており、当該機能を持つ専用チップ、専用回路、または当該機能をコンピュータに果たさせるためのソフトウェア(ツールとしてのソフトウェアも含む)、或いは該ソフトウェアを記録した記録媒体、当該記録媒体を搭載した処理装置・管理装置・ツールとして実現される。

[0268] 情報記憶部2070は、プログラム情報のほか、各種制御情報、目的プログラム等の格納のほか、一時記憶のためのメモリとしても用いられる。

[0269] 図213は、上記の構成を備えるプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置のいずれかとして実施される本発明の動作を示したフローチャートである。

[0270] 同図に示すように、まず、開発対象のソフトウェアに関する要件が、要件をとらえる最小単位である主語＝[名詞, 狭義の属性, 広義の属性]で定義される(ステップS-201)。

[0271] すると、PRD(プロセス・ルート・ダイアグラム)部2030は、名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉える(ステップS-202)。

[0272] 次に、ベクトル形成部2040は、定義部で定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム):  
L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E  
を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで該ベクトルをサブルーチン・プログラムとして形成する(ステップS-203)。

[0273] 次に、パレット部2050は、サブルーチン・プログラムとして形成したベクトルを、基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

に集合させる(ステップS-204)。

[0274] 最後に、パレット連鎖関数部2060は、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数を用いてパレットをプログラム化する(ステップS-205)。

[0275] よって、上記のような構成を備える本発明によれば、本発明の独自の体系に基づいて要求定義をとらえ、それを本発明独自の構造を備えるベクトル(サブルーチン)の未定義部分に代入するので、所望のソフトウェアが、属人性なく得られる。

[0276] 本発明はさらに、上述の「開発対象のソフトウェアを生産する方法」によって生産されたソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現されるが、この場合の本発明は、予め規定された14種類のベクトル(サブルーチン・プログラム)：

L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E

が、ベクトルの実行を管理するパレット関数を伴う3つのパレット：

(01)  $W04i = \Phi 4[\{L4\} + \{O4\} + \{S4\} + R4]i$

(02)  $W02i = \Phi 2[\{L2\} + \{I2\} + R2C + R2]i$

(03)  $W03i = \Phi 3[\{L3\} + R3R + R3C + R3D + R3M + R3E]i$

に集合(パレット化)されて得られる全体に、基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数が配置される構造を有するように構成することもできる。

[0277] またさらに本発明は、上述の「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現されるが、この場合のうちの14種類のベクトル(サブルーチン・プログラム)いずれかとしての本発明は、

L4(論理要素)：未来の自己を表す名詞に真偽値を決定する為の機能を有する；

L2(論理要素)：子孫を有しない自己を表す名詞に真偽値を定義する為の機能を有する；

L3(論理要素)：過去の自己を表す名詞に真偽値を決定する為の機能を有する；

I2(入力作用要素)：上記L2の成立を支援する為の機能を有する；

O4(出力作用要素):上記L4の結果を宣言する為の機能を有する;

S4(同期作用要素):ベクトルの領域を用いてSFの周期性を整備する為の機能を有する;

R4(経路作用要素):W04とW02に順序性を与える為の機能を有する;

R2C(経路作用要素):W02と隣下位のW04に順序性を与える為の機能を有する;

R2(経路作用要素):W02とW03に順序性を与える為の機能を有する;

R3R(経路作用要素):W03とW04に順序性を与える為の機能を有する;

R3C(経路作用要素):W03と隣下位のW04に順序性を与える為の機能を有する;

R3D(経路作用要素):W03と隣上位のW03に順序性を与える為の機能を有する;

R3M(経路作用要素):W03と隣々上位のW04に順序性を与える為の機能を有する;

R3E(経路作用要素):SFの作用を終了させる為の機能を有する;

のいずれかであるように構成してよい。

[0278] また、上記「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現される場合のうちのベクトルの実行管理を行うパレット関数(サブルーチン・プログラム)としての本発明は、

$$(01)W04i = \Phi 4[\{L4\} + \{O4\} + \{S4\} + R4]i]$$

$$(02)W02i = \Phi 2[\{L2\} + \{I2\} + R2C + R2]i]$$

$$(03)W03i = \Phi 3[\{L3\} + R3R + R3C + R3D + R3M + R3E]i]$$

のいずれかの集合化と其の網羅とを行う $\Phi 4$ 、 $\Phi 2$ 、 $\Phi 3$ のいずれかの機能を構造として具備してよい。

[0279] 上記「開発対象のソフトウェアを生産する方法」によってソフトウェアを生産するために用いられるソフトウェアコードの雛型としてのソフトウェア、及び当該ソフトウェアが搭載された記録媒体或いは当該ソフトウェアが搭載された装置(ハードウェア)としても実現される場合のうちのパレット関数の実行管理を行うパレット連鎖関数(プログラム)としての本発明は、3種のパレット:

$$(01)W04i = \Phi 4[\{\{L4\} + \{O4\} + \{S4\} + R4\}i]$$

$$(02)W02i = \Phi 2[\{\{L2\} + \{I2\} + R2C + R2\}i]$$

$$(03)W03i = \Phi 3[\{\{L3\} + R3R + R3C + R3D + R3M + R3E\}i]$$

で構成されるSFに対して、

$$SF_i = \Phi 0[W04 + W02 + W03]i$$

で規定される $\Phi 0$ の構造を具備してよい。

- [0280] さらに、本発明は、上述の「開発対象のソフトウェアを生産する方法」による、要件から抽出した情報(ドキュメント(紙、データ))たる名詞情報、入出力定義体情報、PRD情報のいずれかの情報の抽出方法として、またかかる抽出方法によって抽出された情報(ドキュメント(紙、データ))として、さらには当該抽出された情報の使用方法として、或いは、これらの情報が搭載された情報記録媒体として、または情報の抽出方法／使用方法がコード化されたソフトウェア、当該ソフトウェアが搭載された記録媒体／装置(ハードウェア)として、いずれも実現することができるが、この場合の本発明は、要件から抽出すべき情報として、
- 名詞情報:名詞=[名称, 識別子, 所定の22種のいずれかに分類される名詞種別]  
により定義されるべき情報;
- 入出力定義体情報:入力定義体、出力定義体、並びにそれらに属する名詞の関係に関する情報;
- PRD情報:所定の規則によってSFを順序づける情報を備えて良い。

- [0281] さらに本願発明は、その技術思想の同一及び等価に及ぶ範囲において様々な変形、追加、置換、拡大、縮小等を許容するものである。また、本願発明を用いて生産される装置、方法、ソフトウェア、システムが、その2次的生産品に登載されて商品化された場合であっても、本願発明の価値は何ら減ずるものではない。

- [0282] [表1]

表1: ベクトルの決定規則

IAK: 入力アクセスキー、ICK: 入力処理条件キー、OAK: 出力アクセスキー、OCK: 出力処理条件キー

ベクトル番号	主語										名詞										S 4 の名詞											
	名詞										論理体										S 4 の名詞											
	入力	出力	等価	配列	境界	正規	単語	単語	単語	単語	入力	出力	入力	出力	入力	出力	入力	出力	入力	出力	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3		
1	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
2	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
3	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
4	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
5	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
6	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
7	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
8	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
9	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
10	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
11	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
12	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
13	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
14	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
15	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
16	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
17	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
18	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
19	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
20	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
21	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
22	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
23	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
24	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
25	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
26	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
27	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
28	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
29	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
31	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
32	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

\* 表中の番号は、図31-68の定義規則番号を示す。

[0283] [表2]



表 2：名詞の種別

項番	名詞		定義
1	正規		要件を構成する論理体に属す項目の名称
2	K		1 個の関数関係式で、従属変数と独立変数が同じになる名称
3	M		独立変数となる名称
4	入力 コネクタ	アクセスキー	論理体を代表する名称（複数）
5		処理条件キー	論理体を操作する前提条件を代表する名称（複数）
6		論理体	単位化された名称の集合の名称
7	出力 コネクタ	アクセスキー	論理体を代表する名称（複数）
8		処理条件キー	論理体を操作する前提条件を代表する名称（複数）
9		論理体	単位化された名称の集合の名称
10	パレット	W04	SFを構成するベクトルの3種の器のひとつの名称
11		W02	SFを構成するベクトルの3種の器のひとつの名称
12		W03	SFを構成するベクトルの3種の器のひとつの名称
13	S4（10種類）		SFを同期化する領域名称

[0284] [表3]

表 3：名詞の狭義の属性

項番	属性	定義
1	入力	入力論理体に属す名詞の属性は、狭義の「入力」と定義する。
2	出力	出力論理体に属す名詞の属性は、狭義の「出力」と定義する。
3	配列	1個の名詞が複数の領域を有する場合、その属性は狭義の「配列」と定義する。
4	等価	1個の名詞が複数の論理性を有する場合、その属性は狭義の「等価」と定義する。
5	境界	異なるSFを同期させる名詞の論理性が成立すれば、その属性は狭義の「境界」と定義する。

[0285] [表4]

表 4 ・ ベクトルの原型

項番	種別	所属			分類	役割	ベクトル名		工学的概念
		W04	W02	W03					
1	L4	○			1	論理	論理要素		計算式
2	L2		○		1				付与情報
3	L3			○	1				計算条件
4	I2		○		2	作用	作用要素	入力	読み込み
5	O4	○			2			出力	書き込み
6	S4	○			3		同期作用要素		領域のクリア
7	R4	○			3		経路作用要素	継続	処理の連系
8	R2C		○		1			継続	
9	R2		○		3			継続	
10	R3R			○	3			再起	
11	R3E			○	1			終了	
12	R3C			○	3			継続	
13	R3D			○	3			多重	
14	R3M			○	3			重複	

[0286] [表5]

表 5 : S 4 ベクトルの主語の種別

項番	ΦP	S 4 の主語		
		ベクトル種別	領域種別	
1	W04	L4	第4	
2		O4	第4	
3	W02	L2	第4	
4		I2	第2	
5			第4	
6	W03	L3	入力アクセスキー	第4
7			入力処理条件キー	第4
8			出力アクセスキー	第4
9			出力処理条件キー	第4
10			単語	第4

\*表記されている入出力アクセスキーの領域、入出力処理条件キー、I2、O4の領域を制御BOXと呼ぶ。

[0287] [表6]

表6：ベクトルの原型と領域の関係

項番	主語				Vector								
	種別	属性			第2領域 個数	第4領域 個数	Vector 個数	第2領域 個数	第4領域 個数	Vector 個数	第2領域 個数	第4領域 個数	Vector 個数
					E2			E3			E4		
1	正規				1	1	1	1	1	1	1	1	1
2		○			m	m	1	m	m	1	m	m	1
3			○		—	—	—	1	1	1	1	共通	$n \geq 2$
4		○	○		—	—	—	m	m	1	m	m, 共通	$n \geq 2$
5				○	—	—	—	1	1	1	1	2	1
6			○	○	—	—	—	1	1	1	1	2, 共通	$n \geq 2$
7		○		○	—	—	—	m	m	1	m	2, m	1
8		○	○	○	—	—	—	m	m	1	m	2(m・共通)	$n \geq 2$
9	K				—	1	—	1	1	1	1	1	1
10			○		—	1	—	1	1	1	1	共通	$n \geq 2$
11		○			—	m	—	m	m	1	m	m	1
12	M	○	○		—	m	—	m	m	1	m	m, 共通	$n \geq 2$
13					—	1	—	—	—	—	1	1	1
14			○		—	1	—	—	—	—	1	共通	$n \geq 2$
15		○			—	m	—	—	—	—	m	m	1
16		○	○		—	m	—	—	—	—	m	m, 共通	$n \geq 2$
17	入力論理体: IAK, ICK				—	—	—	1	1	1	—	—	—
18	出力論理体: OAK, OCK				—	—	—	1	1	1	—	—	—
					E2			E4			R系		
19	入力論理体				1	1	1	—	—	—	—	—	—
20	出力論理体				—	—	—	(=L4第4)	1	1	—	—	—
21	経路	R4			—	—	—	—	—	—	1	共通	1
22		R2C			—	—	—	—	—	—	1	共通	1
23		R2			—	—	—	—	—	—	1	共通	1
24		R3R			—	—	—	—	—	—	1	共通	1
25		R3E			—	—	—	—	—	—	1	共通	1

\* IAK：入力アクセスキー、ICK：入力処理条件キーを示す。  
\* OAK：出力アクセスキー、OCK：出力処理条件キーを示す。  
\* nは等価性を有する単語の論理性の種別数を示す。  
\* mは配列を有する単語の配列定数（列と行の積数）を示す。  
\* L4第4領域の2は、正規と境界の2つの領域が存在する事を示す。

[0288] [表7]

表7：名詞の広義の属性

項番	名詞	定義
1	座標	S F（またはPRD）におけるパレットの位置情報。
2	ベクトル種別	表4参照。
3	第2規約	W02 W02に属すベクトルの第2規約の論理性。
4		W03 W03に属すベクトルの第2規約の論理性。
5		W04 W04に属すベクトルの第2規約の論理性。
6	ベクトル領域	4種 ベクトルを構成する領域の個数。
7		型 名詞の文字タイプ
8		桁数 名詞の桁数
9		小数桁数 名詞の小数桁
10		配列 名詞の配列（表3）

[0289] [表8]

表 8 : 経路制御情報テーブル

項番	自SF情報		経路識別子	行き先ΦP-ID				作用R3C情報への 経路発報記録(n個有)		04識別子 (n個有)	04成立 情報	SF終了 情報
	SF-ID	ΦP-ID		隣々上位 Φ4-ID	隣上位 Φ3-ID	隣下位 Φ4-ID	発報 記録					
1	1, 1	W03-1, 1	R3R-1	—	—	—	—	—	—	—	—	
			R3E	—	—	—	—	—	—	—	—	
		W04-1, 1	—	—	—	—	—	—	—	WT01		
2	2, 1	W03-2, 1	R3R-2	—	—	—						
			R3D-2	—	W03-1, 1	—						
			R3C-21	—	—	W04-3, 1						
			R3E	—	—	—	—	—	—	—	—	—
		W04-2, 1	—	—	—	—	—	—	—	WT02		
3	3, 1	W03-3, 1	R3R-3	—	—	—						
			R3D-3	—	W03-2, 1	—						
			R3M-3	W04-1, 1	—	—		SF(2, 1) R3C				
			R3E	—	—	—	—	—	—	—	—	
		W04-3, 1	—	—	—	—	—	—	—	WT03		
備考	PRD情報から生成							Φ0で更新		PRD情報 から生成	Φ0で更新	Φ3のR3E で更新

\*SF(i, j)は、PRD (図 2 4 参照) におけるSFの座標。  
\*ΦP-IDは、パレット関数の識別子。  
\*経路発報記録は、パレット連鎖関数に組込まれる専用のプログラムで制御される。  
\*本表は、図 2 4 のPRDから定義されるものである。

[0290] [表9]



表9：開発実績一覧表

項番	業種	システム名	規模			言語	システム構成
			画面	帳票	ページ		
1	流通業	販売管理システム	543	297	1	COBOL、RPG	AS400ホスト、DB2
2		販売顧客管理システム	192	197	1	C	WindowsNT、C/S、ORACLE
3		固定資産システム	37	17	1	COBOL	AS400ホスト、DB2
4		ストアネットシステムNT版	247	104	48	C	WindowsNT、C/S、ORACLE
5		ストアネットシステムAS版	232	95	28	C	AS400ホスト、C/S、DB2
6		eコマースシステム	30	4	1	Java	RS/6000 AIX V4.3.3、IBM HTTP Server 1.3.6.2、WebSphereV3.54、DB2 UDB V7.1
7		トレビスシステム	135	26	61	VB	WindowsNT4.0、MS IIS Web、ORACLE
-	大手鉄鋼	販売/マスタ/経営/顧客管理システム	92	658	1	Java、RPG	AS400、WebSphere、AS400 RDB
8		予算品種別原価管理システム	105	1	146	COBOL	IBMホスト、C/S、DB2
9		バーチャルマーケットシステム	86	1	1	Java	Linux(Red Hat 7.2Deluxe)、Tomcat4、DB2、Apache2
10	大手保険	輸送品管理システム(受発注管理)	143	46	1	COBOL	IBMホスト、C/S、ORACLE
11		4.0 TK運営管理機関システム	672	204	479	COBOL	Sun Solaris、TPI/Web、ORACLE
12		新幹線品管理システム(受発注管理)	295	139	67	COBOL	Sun Solaris、TPI/Web、ORACLE
13	大手銀行	有価証券管理システム	46	38	1	VB	Windows2000、PCx7277、ACCESS
14		預貸残高管理システム	1	1	67	PL/1	IBMホスト、IBM-2860、IMS
15	大手重工業	新資材管理システム#1	64	2	1	Java	WindowsNT、WebSphere、SQL Server
16		新資材管理システム#2	51	4	1	Java	WindowsNT、WebSphere、SQL Server
17		生産管理システム#1	28	1	1	Java	WindowsNT、WebSphere、SQL Server
18		生産管理システム#2	63	1	1	Java	WindowsNT、WebSphere、SQL Server
19		生産管理システム#3	13	1	1	Java	WindowsNT、WebSphere、SQL Server
20		支給品管理システム	8	13	1	Java	WindowsNT、WebSphere、SQL Server
21		定期修理部品管理システム	19	37	61	Java	WindowsNT、WebSphere、SQL Server
22	金融系関連	料金計算システム	46	2	60	VB	WindowsNTx7277、DB2
23	大手建設	業務管理システム	150	24	1	Java、VB	Windows2000、WebSphere、DB2
24		有価証券システム	40	40	10	Java	Solaris、WebLogic、SuperVisualFormade、DB2
25	大手パナソニック	債権管理システム	56	20	1	VB	Windows2000、C/S、SQL Server7.0、ActiveReport、Excel
26	S.I.企業	開発作業管理システム	5	1	1	Java	MiracleLinux、OracleApplicationServer、SkreenMill、OracleDB
27		開発作業管理&実績分析システム	12	1	1	VB	WindowsNT、MSIIS、ASP、SQL Server
28		営業関連状況照会システム	7	1	1	Java	Solaris、OracleApplicationServer、JSP、OracleDB
29		教育演習課題	1	1	8	COBOL97	Windows2000PCx7277
30		本部業務管理パッチシステム	1	1	30	COBOL	IBMホスト

\*本表は、2002年までの実績一覧表である。

## 資料1：ベクトルの定義規則

定義規則1：名詞が正規で狭義の属性が入力のベクトル

定義規則2：名詞が正規で狭義の属性が入力、配列のベクトル

定義規則3：名詞が正規で狭義の属性が出力のベクトル

定義規則4：名詞が正規で狭義の属性が出力、等価のベクトル

定義規則5：名詞が正規で狭義の属性が出力、配列のベクトル



- 定義規則6 :名詞が正規で狭義の属性が出力, 等価, 配列のベクトル
- 定義規則7 :名詞が正規で狭義の属性が出力, 境界のベクトル
- 定義規則8 :名詞が正規で狭義の属性が出力, 等価, 境界のベクトル
- 定義規則9 :名詞が正規で狭義の属性が出力, 配列, 境界のベクトル
- 定義規則10:名詞が正規で狭義の属性が出力, 等価, 配列, 境界のベクトル
- 定義規則11:名詞がKのベクトル
- 定義規則12:名詞がKで狭義の属性が等価のベクトル
- 定義規則13:名詞がKで狭義の属性が配列のベクトル
- 定義規則14:名詞がKで狭義の属性が等価, 配列のベクトル
- 定義規則15:名詞がMで狭義の属性が出力のベクトル
- 定義規則16:名詞がMで狭義の属性が出力, 等価のベクトル
- 定義規則17:名詞がMで狭義の属性が出力, 配列のベクトル
- 定義規則18:名詞がMで狭義の属性が出力, 等価, 配列のベクトル
- 定義規則19:名詞が論理体で狭義の属性が入力のベクトル
- 定義規則20:名詞がアクセスキーで狭義の属性が出力のベクトル
- 定義規則21:名詞が処理条件キーで狭義の属性が出力のベクトル
- 定義規則22:名詞が論理体で狭義の属性が出力のベクトル
- 定義規則23:名詞がアクセスキーで狭義の属性が出力のベクトル
- 定義規則24:名詞が処理条件キーで狭義の属性が出力のベクトル
- 定義規則25:名詞がパレットW04でR4のベクトル
- 定義規則26:名詞がパレットW02でR2Cのベクトル
- 定義規則27:名詞がパレットW02でR2のベクトル
- 定義規則28:名詞がパレットW03でR3Rのベクトル
- 定義規則29:名詞がパレットW03でR3Eのベクトル
- 定義規則30:名詞がパレットW03でR3Cのベクトル
- 定義規則31:名詞がパレットW03でR3Dのベクトル
- 定義規則32:名詞がパレットW03でR3Mのベクトル
- 定義規則33:名詞がI2第2領域のS4のベクトル

定義規則34:名詞がI2第4領域のS4のベクトル

定義規則35:名詞が入力アクセスキーのL3第4領域のS4のベクトル

定義規則36:名詞が入力処理条件キーのL3第4領域のS4のベクトル

定義規則37:名詞がO4第4領域のS4のベクトル

定義規則38:名詞が出力アクセスキーのL3第4領域のS4のベクトル

定義規則39:名詞が出力処理条件キーのL3第4領域のS4のベクトル

定義規則40:名詞がL4第4領域のS4のベクトル

定義規則41から51については定義規則40に準じる。

定義規則52:名詞がL2第4領域のS4のベクトル

定義規則53から57については定義規則52に準じる。

定義規則58:名詞がL3第4領域のS4のベクトル

定義規則59から73については定義規則58に準じる。

資料2:テンプレート(プログラム言語がVBの例)

定義規則1で定義されるベクトルL2(正規)

定義規則2で定義されるベクトルL2(正規)

定義規則3で定義されるベクトルL3(正規)

定義規則3で定義されるベクトルL4(正規)

定義規則4で定義されるベクトルL3(正規)

定義規則4で定義されるベクトルL4(正規)

定義規則5で定義されるベクトルL3(正規)

定義規則5で定義されるベクトルL4(正規)

定義規則6で定義されるベクトルL3(正規)

定義規則6で定義されるベクトルL4(正規)

定義規則7で定義されるベクトルL3(正規)

定義規則7で定義されるベクトルL4(正規)

定義規則8で定義されるベクトルL3(正規)

定義規則8で定義されるベクトルL4(正規)

定義規則9で定義されるベクトルL3(正規)

定義規則9で定義されるベクトルL4(正規)

定義規則10で定義されるベクトルL3(正規)

定義規則10で定義されるベクトルL4(正規)

定義規則11で定義されるベクトルL3(K)

定義規則11で定義されるベクトルL4(K)

定義規則15で定義されるベクトルL4(M)については上記参照

定義規則16で定義されるベクトルL4(M)については上記参照

定義規則11で定義されるベクトルL2(K)

定義規則12で定義されるベクトルL2(K)については上記参照

定義規則15で定義されるベクトルL2(M)については上記参照

定義規則16で定義されるベクトルL2(M)については上記参照

定義規則11で定義されるベクトルL3(Kの派生元の正規)

定義規則12で定義されるベクトルL3(Kの派生元の正規)については上記参照

定義規則11で定義されるベクトルL4(Kの派生元の正規)

定義規則12で定義されるベクトルL4(Kの派生元の正規)については上記参照

定義規則12で定義されるベクトルL3(K)

定義規則12で定義されるベクトルL4(K)

定義規則13で定義されるベクトルL3(K)

定義規則13で定義されるベクトルL4(K)

定義規則17で定義されるベクトルL4(M)については上記参照

定義規則18で定義されるベクトルL4(M)については上記参照

定義規則13で定義されるベクトルL2(K)

定義規則14で定義されるベクトルL2(K)については上記参照

定義規則17で定義されるベクトルL2(M)については上記参照

定義規則18で定義されるベクトルL2(M)については上記参照

定義規則13で定義されるベクトルL3(Kの派生元の正規)

定義規則14で定義されるベクトルL3(Kの派生元の正規)については上記参照

定義規則13で定義されるベクトルL4(Kの派生元の正規)

定義規則14で定義されるベクトルL4(Kの派生元の正規)については上記参照

定義規則14で定義されるベクトルL3(K)

定義規則14で定義されるベクトルL4(K)

定義規則19で定義されるベクトルI2(入力論理体)

定義規則20で定義されるベクトルL3(入力アクセスキー)

定義規則21で定義されるベクトルL3(入力処理条件キー)

定義規則22で定義されるベクトルO4(出力論理体)

定義規則23で定義されるベクトルL3(出力アクセスキー)

定義規則24で定義されるベクトルL3(出力処理条件キー)

定義規則25で定義されるベクトルR4(W04パレット)

定義規則26で定義されるベクトルR2C(W02パレット)

定義規則29で定義されるベクトルR3E(W03パレット)については上記参照

定義規則30で定義されるベクトルR3C(W03パレット)については上記参照

定義規則31で定義されるベクトルR3D(W03パレット)については上記参照

定義規則32で定義されるベクトルR3M(W03パレット)については上記参照

定義規則27で定義されるベクトルR2(W02パレット)

定義規則28で定義されるベクトルR3R(W03パレット)

定義規則33で定義されるベクトルS4

定義規則34で定義されるベクトルS4

定義規則35で定義されるベクトルS4

定義規則36で定義されるベクトルS4については上記参照

定義規則37で定義されるベクトルS4

定義規則38で定義されるベクトルS4

定義規則39で定義されるベクトルS4については上記参照

定義規則40で定義されるベクトルS4

定義規則41で定義されるベクトルS4

定義規則42で定義されるベクトルS4

定義規則43で定義されるベクトルS4

定義規則44で定義されるベクトルS4

定義規則45で定義されるベクトルS4

定義規則46で定義されるベクトルS4

定義規則47で定義されるベクトルS4

定義規則48で定義されるベクトルS4

定義規則49で定義されるベクトルS4

定義規則50で定義されるベクトルS4

定義規則51で定義されるベクトルS4

定義規則52で定義されるベクトルS4

定義規則54で定義されるベクトルS4については上記参照

定義規則55で定義されるベクトルS4については上記参照

定義規則53で定義されるベクトルS4

定義規則56で定義されるベクトルS4については上記参照

定義規則57で定義されるベクトルS4については上記参照

定義規則58で定義されるベクトルS4

定義規則59で定義されるベクトルS4

定義規則60で定義されるベクトルS4

定義規則61で定義されるベクトルS4

定義規則62で定義されるベクトルS4

定義規則63で定義されるベクトルS4

定義規則64で定義されるベクトルS4

定義規則65で定義されるベクトルS4

定義規則66で定義されるベクトルS4

定義規則67で定義されるベクトルS4

定義規則68で定義されるベクトルS4

定義規則69で定義されるベクトルS4

定義規則70で定義されるベクトルS4



定義規則71で定義されるベクトルS4

定義規則72で定義されるベクトルS4

定義規則73で定義されるベクトルS4

産業上の利用可能性

[0291] 上記で詳細に説明したように、本発明はソフトウェア産業のみならず、ソフトウェアを利用する全産業に極めて有用な効果をもたらす。

図面の簡単な説明

[0292] [図1]本発明の一実施形態に係るIDSの模式図である。

[図2]本発明の一実施形態に係る論理原子( $V_i$ )の空間速度を表す概念図である。

[図3]本発明の一実施形態に係る占有速度の概念を説明するための概念図である。

[図4]本発明の一実施形態に係る空の連鎖の概念を説明するための概念図である。

[図5]本発明の一実施形態に係る単元の連鎖の概念を説明するための概念図である。

[図6]本発明の一実施形態に係る連糸の概念を説明するための概念図である。

[図7]本発明の一実施形態に係る臨界連鎖の概念を説明するための概念図である。

[図8]本発明の一実施形態に係る特異連鎖の概念を説明するための概念図である。

[図9]本発明の一実施形態に係る意図の概念を説明するための概念図である。

[図10]本発明の一実施形態に係る意識の概念を説明するための概念図である。

[図11]本発明の一実施形態に係る自己なる存在のモデルを概念的に説明するための図である。

[図12]本発明の一実施形態に係る脈絡のモデルを概念的に説明するための図である。

[図13]本発明の一実施形態に係る自己の存在を証明する方法を概念的に説明するための図である。

[図14]本発明の一実施形態に係る自己の未来の言表構造(PS4)を説明するための図である。

[図15]本発明の一実施形態に係る過去でも未来でもない自己の言表構造(PS2)を説明するための図である。

[図16]本発明の一実施形態に係る過去の自己の言表構造(PS3)を説明するための図である。

[図17]本発明の一実施形態に係る未来の自己が存在しない自己の言表構造(PS2<sup>1</sup>)を説明するための図である。

[図18]本発明の一実施形態に係るENWのモデルを説明するための図である。

[図19]本発明の一実施形態に係るTDMのモデルを説明するための図である。

[図20]本発明の一実施形態に係る要件の捉え方を説明するための図である。

[図21]本発明の一実施形態に係るプログラムの位置付けを説明するための図である。

[図22]本発明の一実施形態に係るベクトルの標準形を説明するための図である。

[図23]本発明の一実施形態に係るSF(基本構造)を概念的に説明するための図である。

[図24]本発明の一実施形態に係るPRDを概念的に説明するための図である。

[図25]本発明の一実施形態に係るパレット連鎖関数の標準形を概念的に説明するための図である。

[図26]本発明の一実施形態に係るパレット連鎖関数の標準形を概念的に説明するための図である。

[図27]本発明の一実施形態に係るパレット関数の標準形を概念的に説明するための図である。

[図28]従来のプログラム構築のアプローチを概念的に説明するための図である。

[図29]本発明の要件を捉える公式を概念的に説明するための図である。

[図30]本発明のアプローチを概念的に説明するための図である。

[図31]本発明の一実施形態に係る定義規則1:名詞が正規で狭義の属性が入力のベクトルを概念的に説明するための図である。

[図32]本発明の一実施形態に係る定義規則2:名詞が正規で狭義の属性が入力, 配列のベクトルを概念的に説明するための図である。

[図33]本発明の一実施形態に係る定義規則3:名詞が正規で狭義の属性が出力のベクトルを概念的に説明するための図である。

[図34]本発明の一実施形態に係る定義規則4:名詞が正規で狭義の属性が出力, 等価のベクトルを概念的に説明するための図である。

[図35]本発明の一実施形態に係る定義規則5:名詞が正規で狭義の属性が出力, 配列のベクトルを概念的に説明するための図である。

[図36]本発明の一実施形態に係る定義規則6:名詞が正規で狭義の属性が出力, 等価, 配列のベクトルを概念的に説明するための図である。

[図37]本発明の一実施形態に係る定義規則7:名詞が正規で狭義の属性が出力, 境界のベクトルを概念的に説明するための図である。

[図38]本発明の一実施形態に係る定義規則8:名詞が正規で狭義の属性が出力, 等価, 境界のベクトルを概念的に説明するための図である。

[図39]本発明の一実施形態に係る定義規則9:名詞が正規で狭義の属性が出力, 配列, 境界のベクトルを概念的に説明するための図である。

[図40]本発明の一実施形態に係る定義規則10:名詞が正規で狭義の属性が出力, 等価, 配列, 境界のベクトルを概念的に説明するための図である。

[図41]本発明の一実施形態に係る定義規則11:名詞がKのベクトルを概念的に説明するための図である。

[図42]本発明の一実施形態に係る定義規則12:名詞がKで狭義の属性が等価のベクトルを概念的に説明するための図である。

[図43]本発明の一実施形態に係る定義規則13:名詞がKで狭義の属性が配列のベクトルを概念的に説明するための図である。

[図44]本発明の一実施形態に係る定義規則14:名詞がKで狭義の属性が等価, 配列のベクトルを概念的に説明するための図である。

[図45]本発明の一実施形態に係る定義規則15:名詞がMで狭義の属性が出力のベクトルを概念的に説明するための図である。

[図46]本発明の一実施形態に係る定義規則16:名詞がMで狭義の属性が出力, 等価のベクトルを概念的に説明するための図である。

[図47]本発明の一実施形態に係る定義規則17:名詞がMで狭義の属性が出力, 配列のベクトルを概念的に説明するための図である。

[図48]本発明の一実施形態に係る定義規則18:名詞がMで狭義の属性が出力, 等価, 配列のベクトルを概念的に説明するための図である。

[図49]本発明の一実施形態に係る定義規則19:名詞が論理体で狭義の属性が入力のベクトル、定義規則20:名詞がアクセスキーで狭義の属性が出力のベクトル、及び定義規則21:名詞が処理条件キーで狭義の属性が出力のベクトルを概念的に説明するための図である。

[図50]本発明の一実施形態に係る定義規則22:名詞が論理体で狭義の属性が出力のベクトル、定義規則23:名詞がアクセスキーで狭義の属性が出力のベクトル、及び定義規則24:名詞が処理条件キーで狭義の属性が出力のベクトルを概念的に説明するための図である。

[図51]本発明の一実施形態に係る定義規則25:名詞がパレットW04でR4のベクトルを概念的に説明するための図である。

[図52]本発明の一実施形態に係る定義規則26:名詞がパレットW02でR2Cのベクトルを概念的に説明するための図である。

[図53]本発明の一実施形態に係る定義規則27:名詞がパレットW02でR2のベクトルを概念的に説明するための図である。

[図54]本発明の一実施形態に係る定義規則28:名詞がパレットW03でR3Rのベクトルを概念的に説明するための図である。

[図55]本発明の一実施形態に係る定義規則29:名詞がパレットW03でR3Eのベクトルを概念的に説明するための図である。

[図56]本発明の一実施形態に係る定義規則30:名詞がパレットW03でR3Cのベクトルを概念的に説明するための図である。

[図57]本発明の一実施形態に係る定義規則31:名詞がパレットW03でR3Dのベクトルを概念的に説明するための図である。

[図58]本発明の一実施形態に係る定義規則32:名詞がパレットW03でR3Mのベクトルを概念的に説明するための図である。

[図59]本発明の一実施形態に係る定義規則33:名詞がI2第2領域のS4のベクトルを概念的に説明するための図である。

[図60]本発明の一実施形態に係る定義規則34:名詞がI2第4領域のS4のベクトルを概念的に説明するための図である。

[図61]本発明の一実施形態に係る定義規則35:名詞が入力アクセスキーのL3第4領域のS4のベクトルを概念的に説明するための図である。

[図62]本発明の一実施形態に係る定義規則36:名詞が入力処理条件キーのL3第4領域のS4のベクトルを概念的に説明するための図である。

[図63]本発明の一実施形態に係る定義規則37:名詞がO4第4領域のS4のベクトルを概念的に説明するための図である。

[図64]本発明の一実施形態に係る定義規則38:名詞が出力アクセスキーのL3第4領域のS4のベクトルを概念的に説明するための図である。

[図65]本発明の一実施形態に係る定義規則39:名詞が出力処理条件キーのL3第4領域のS4のベクトルを概念的に説明するための図である。

[図66]本発明の一実施形態に係る定義規則40:名詞がL4第4領域のS4のベクトルを概念的に説明するための図である。

[図67]本発明の一実施形態に係る定義規則52:名詞がL2第4領域のS4のベクトルを概念的に説明するための図である。

[図68]本発明の一実施形態に係る定義規則58:名詞がL3第4領域のS4のベクトルを概念的に説明するための図である。

[図69]本発明の一実施形態に係る定義規則1で定義されるベクトルL2(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図70]本発明の一実施形態に係る定義規則1で定義されるベクトルL2(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図71]本発明の一実施形態に係る定義規則2で定義されるベクトルL2(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図72]本発明の一実施形態に係る定義規則2で定義されるベクトルL2(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図73]本発明の一実施形態に係る定義規則2で定義されるベクトルL2(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。



[図74]本発明の一実施形態に係る定義規則3で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図75]本発明の一実施形態に係る定義規則3で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図76]本発明の一実施形態に係る定義規則3で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図77]本発明の一実施形態に係る定義規則3で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図78]本発明の一実施形態に係る定義規則4で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図79]本発明の一実施形態に係る定義規則4で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図80]本発明の一実施形態に係る定義規則4で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図81]本発明の一実施形態に係る定義規則4で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図82]本発明の一実施形態に係る定義規則5で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図83]本発明の一実施形態に係る定義規則5で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図84]本発明の一実施形態に係る定義規則5で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図85]本発明の一実施形態に係る定義規則5で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図86]本発明の一実施形態に係る定義規則5で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図87]本発明の一実施形態に係る定義規則5で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図88]本発明の一実施形態に係る定義規則5で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図89]本発明の一実施形態に係る定義規則6で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図90]本発明の一実施形態に係る定義規則6で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図91]本発明の一実施形態に係る定義規則6で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図92]本発明の一実施形態に係る定義規則6で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図93]本発明の一実施形態に係る定義規則6で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図94]本発明の一実施形態に係る定義規則6で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図95]本発明の一実施形態に係る定義規則6で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図96]本発明の一実施形態に係る定義規則7で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図97]本発明の一実施形態に係る定義規則7で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図98]本発明の一実施形態に係る定義規則7で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図99]本発明の一実施形態に係る定義規則7で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図100]本発明の一実施形態に係る定義規則8で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図101]本発明の一実施形態に係る定義規則8で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図102]本発明の一実施形態に係る定義規則8で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図103]本発明の一実施形態に係る定義規則8で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図104]本発明の一実施形態に係る定義規則9で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図105]本発明の一実施形態に係る定義規則9で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図106]本発明の一実施形態に係る定義規則9で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図107]本発明の一実施形態に係る定義規則9で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図108]本発明の一実施形態に係る定義規則9で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図109]本発明の一実施形態に係る定義規則9で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図110]本発明の一実施形態に係る定義規則9で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図111]本発明の一実施形態に係る定義規則10で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図112]本発明の一実施形態に係る定義規則10で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図113]本発明の一実施形態に係る定義規則10で定義されるベクトルL3(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図114]本発明の一実施形態に係る定義規則10で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図115]本発明の一実施形態に係る定義規則10で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図116]本発明の一実施形態に係る定義規則10で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図117]本発明の一実施形態に係る定義規則10で定義されるベクトルL4(正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図118]本発明の一実施形態に係る定義規則11で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図119]本発明の一実施形態に係る定義規則11で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図120]本発明の一実施形態に係る定義規則11で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図121]本発明の一実施形態に係る定義規則11で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図122]本発明の一実施形態に係る定義規則11で定義されるベクトルL2(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図123]本発明の一実施形態に係る定義規則11で定義されるベクトルL2(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図124]本発明の一実施形態に係る定義規則11で定義されるベクトルL3(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図125]本発明の一実施形態に係る定義規則11で定義されるベクトルL3(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図126]本発明の一実施形態に係る定義規則11で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図127]本発明の一実施形態に係る定義規則11で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図128]本発明の一実施形態に係る定義規則12で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図129]本発明の一実施形態に係る定義規則12で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。



[図130]本発明の一実施形態に係る定義規則12で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図131]本発明の一実施形態に係る定義規則12で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図132]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図133]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図134]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図135]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図136]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図137]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図138]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図139]本発明の一実施形態に係る定義規則13で定義されるベクトルL2(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図140]本発明の一実施形態に係る定義規則13で定義されるベクトルL2(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図141]本発明の一実施形態に係る定義規則13で定義されるベクトルL2(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図142]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図143]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。



[図144]本発明の一実施形態に係る定義規則13で定義されるベクトルL3(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図145]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図146]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図147]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図148]本発明の一実施形態に係る定義規則13で定義されるベクトルL4(Kの派生元の正規)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図149]本発明の一実施形態に係る定義規則14で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図150]本発明の一実施形態に係る定義規則14で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図151]本発明の一実施形態に係る定義規則14で定義されるベクトルL3(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図152]本発明の一実施形態に係る定義規則14で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図153]本発明の一実施形態に係る定義規則14で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図154]本発明の一実施形態に係る定義規則14で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図155]本発明の一実施形態に係る定義規則14で定義されるベクトルL4(K)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図156]本発明の一実施形態に係る定義規則19で定義されるベクトルI2(入力論理体)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図157]本発明の一実施形態に係る定義規則19で定義されるベクトルI2(入力論理体)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図158]本発明の一実施形態に係る定義規則20で定義されるベクトルL3(入力アクセスキー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図159]本発明の一実施形態に係る定義規則20で定義されるベクトルL3(入力アクセスキー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図160]本発明の一実施形態に係る定義規則21で定義されるベクトルL3(入力処理条件キー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図161]本発明の一実施形態に係る定義規則21で定義されるベクトルL3(入力処理条件キー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図162]本発明の一実施形態に係る定義規則22で定義されるベクトルO4(出力論理体)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図163]本発明の一実施形態に係る定義規則22で定義されるベクトルO4(出力論理体)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図164]本発明の一実施形態に係る定義規則22で定義されるベクトルO4(出力論理体)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図165]本発明の一実施形態に係る定義規則23で定義されるベクトルL3(出力アクセスキー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図166]本発明の一実施形態に係る定義規則23で定義されるベクトルL3(出力アクセスキー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図167]本発明の一実施形態に係る定義規則24で定義されるベクトルL3(出力処理条件キー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図168]本発明の一実施形態に係る定義規則24で定義されるベクトルL3(出力処理条件キー)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図169]本発明の一実施形態に係る定義規則25で定義されるベクトルR4(W04パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図170]本発明の一実施形態に係る定義規則25で定義されるベクトルR4(W04パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図171]本発明の一実施形態に係る定義規則26で定義されるベクトルR2C(W02パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図172]本発明の一実施形態に係る定義規則26で定義されるベクトルR2C(W02パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図173]本発明の一実施形態に係る定義規則27で定義されるベクトルR2(W02パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図174]本発明の一実施形態に係る定義規則27で定義されるベクトルR2(W02パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図175]本発明の一実施形態に係る定義規則28で定義されるベクトルR3R(W03パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図176]本発明の一実施形態に係る定義規則28で定義されるベクトルR3R(W03パレット)のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図177]本発明の一実施形態に係る定義規則33で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図178]本発明の一実施形態に係る定義規則34で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図179]本発明の一実施形態に係る定義規則35で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図180]本発明の一実施形態に係る定義規則37で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図181]本発明の一実施形態に係る定義規則38で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図182]本発明の一実施形態に係る定義規則40で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図183]本発明の一実施形態に係る定義規則41で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図184]本発明の一実施形態に係る定義規則42で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図185]本発明の一実施形態に係る定義規則43で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図186]本発明の一実施形態に係る定義規則44で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図187]本発明の一実施形態に係る定義規則45で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図188]本発明の一実施形態に係る定義規則46で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図189]本発明の一実施形態に係る定義規則47で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図190]本発明の一実施形態に係る定義規則48で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図191]本発明の一実施形態に係る定義規則49で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図192]本発明の一実施形態に係る定義規則50で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図193]本発明の一実施形態に係る定義規則51で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図194]本発明の一実施形態に係る定義規則52で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図195]本発明の一実施形態に係る定義規則53で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図196]本発明の一実施形態に係る定義規則58で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図197]本発明の一実施形態に係る定義規則59で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図198]本発明の一実施形態に係る定義規則60で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図199]本発明の一実施形態に係る定義規則61で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。



[図200]本発明の一実施形態に係る定義規則62で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図201]本発明の一実施形態に係る定義規則63で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図202]本発明の一実施形態に係る定義規則64で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図203]本発明の一実施形態に係る定義規則65で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図204]本発明の一実施形態に係る定義規則66で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図205]本発明の一実施形態に係る定義規則67で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図206]本発明の一実施形態に係る定義規則68で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図207]本発明の一実施形態に係る定義規則69で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図208]本発明の一実施形態に係る定義規則70で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図209]本発明の一実施形態に係る定義規則71で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図210]本発明の一実施形態に係る定義規則72で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図211]本発明の一実施形態に係る定義規則73で定義されるベクトルS4のテンプレート(プログラム言語がVBの例)に係るコードの一例である。

[図212]本発明の一実施形態に係るプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置の構成を示した機能ブロック図である。



[図213]本発明の一実施形態に係る上記の構成を備えるプログラム(ソフトウェア)、プログラム生成装置、プログラム処理装置、ツール(装置として或いはソフトウェアとしての双方を含む)、ソフトウェア開発装置、ソフトウェア開発支援装置、或いはソフトウェア開発管理装置の動作を示したフローチャートである。

## 符号の説明

[0293] PRD 処理経路図

SF シナリオ関数

2010 全体制御部

2030 PRD(プロセス・ルート・ダイアグラム)部

2040 ベクトル形成部

2050 パレット部

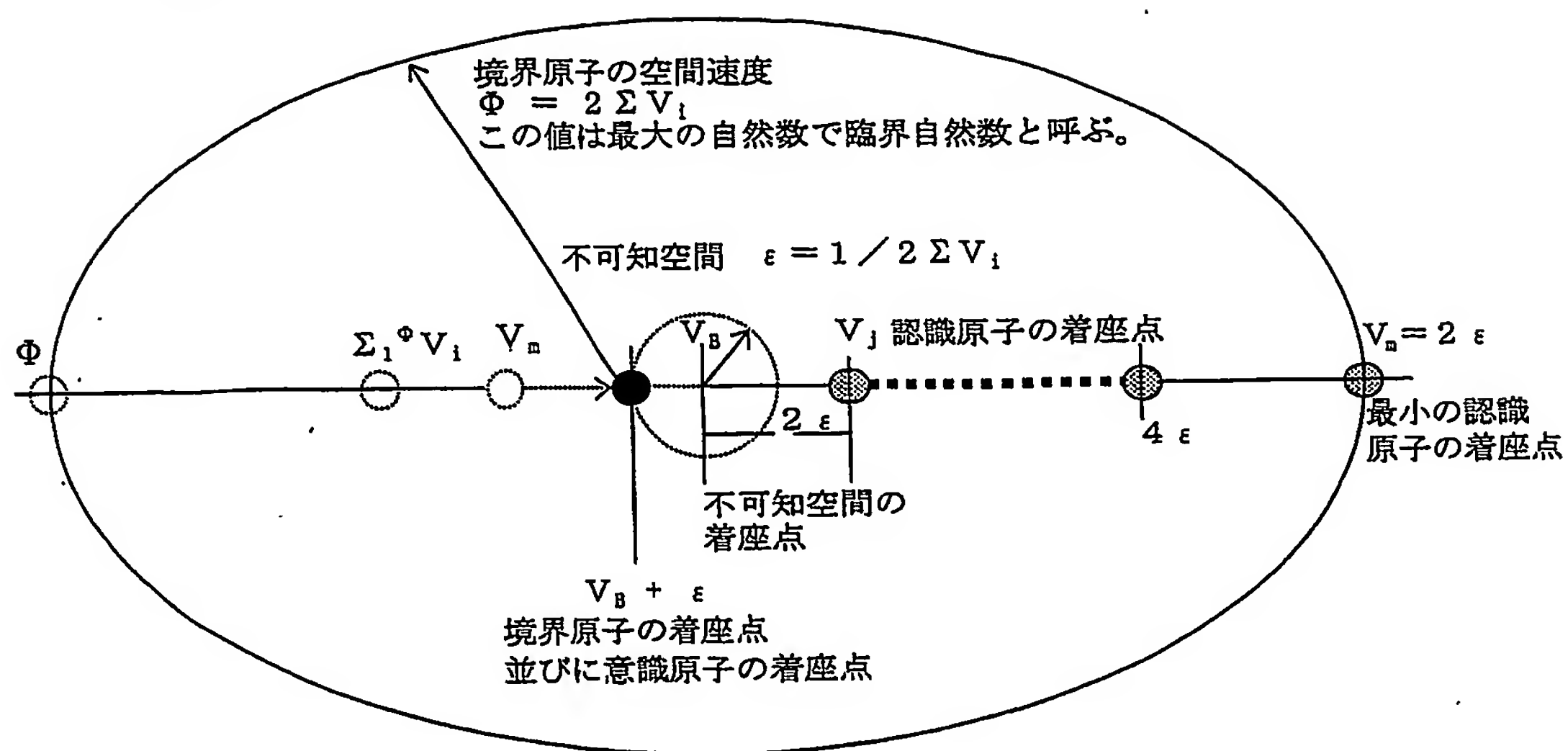
2060 パレット連鎖関数部

2070 情報記憶部

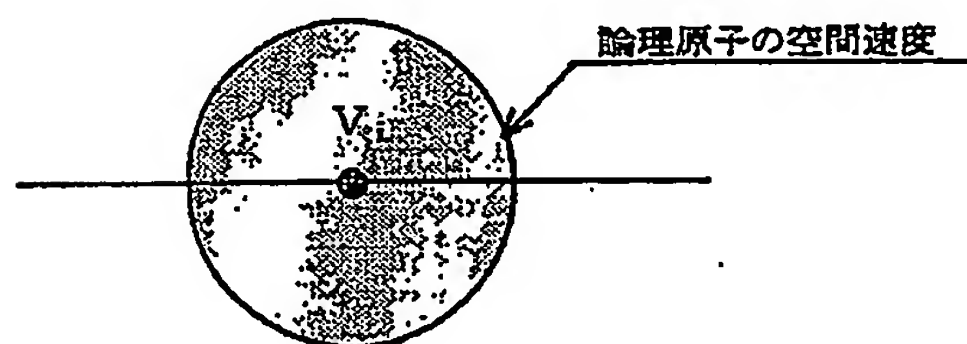
## 請求の範囲

- [1] 開発対象のソフトウェアに関する要件を、要件をとらえる最小単位である主語＝[名詞, 狭義の属性, 広義の属性]で定義する第1のステップと、
- 名詞が属する入力定義体、出力定義体をとらえる構造である基本構造(SF)、並びに、これら基本構造の関係と成立順序を所定の規則に従って、PRD(プロセス・ルート・ダイアグラム)として捉える第2のステップと、
- 前記第1のステップで定義された名詞の属性から、各名詞に該当する予め規定された14種類のベクトル(サブルーチン・プログラム)：
- L4、L2、L3、I2、O4、S4、R4、R2C、R2、R3R、R3C、R3D、R3M、R3E
- を決定し、各ベクトルの未定義部分に該当する名詞の情報を埋め込み、未定義部分を埋め込むことで前記ベクトルをサブルーチン・プログラムとして形成する第3のステップと、
- 前記サブルーチン・プログラムとして形成したベクトルを、前記基本構造ごとに、ベクトルの実行を管理するパレット関数を伴う3つのパレット：
- (01)  $W04i = \Phi 4[\{L4\} + \{O4\} + \{S4\} + R4]i$
- (02)  $W02i = \Phi 2[\{L2\} + \{I2\} + R2C + R2]i$
- (03)  $W03i = \Phi 3[\{L3\} + R3R + R3C + R3D + R3M + R3E]i$
- に集合させる第4のステップと、
- 基本構造ごとに集合されたパレットの実行を管理するパレット連鎖関数を用いて前記パレットをプログラム化する第5のステップと
- を具備することを特徴とするソフトウェアの生産方法。

[図1]

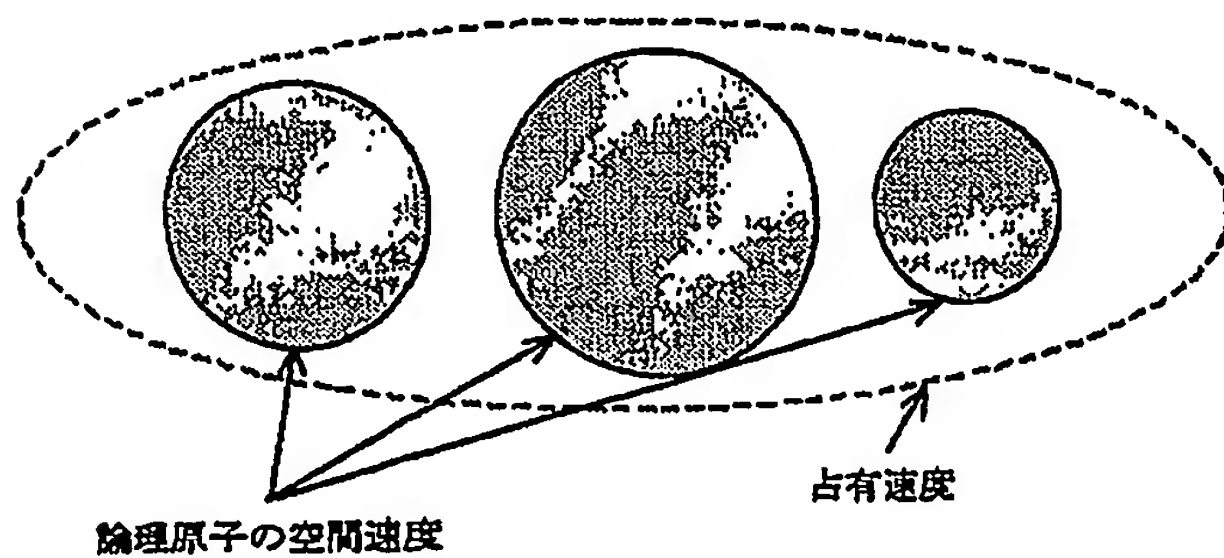


[図2]

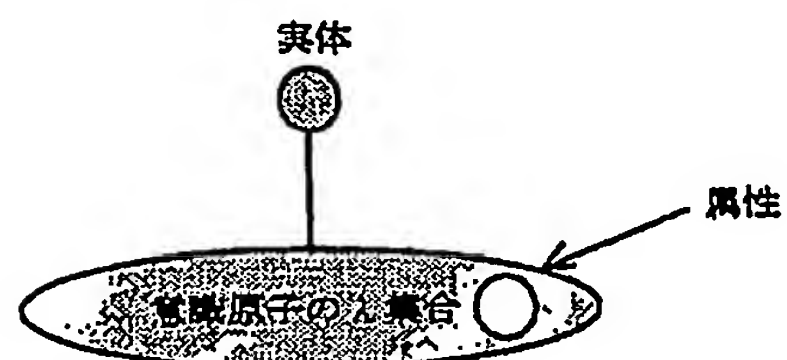


論理原子  $V_i$  の空間速度の定義  $= 1 / |V_i - V_B|$   
 $V_B$  : 境界原子の時間速度

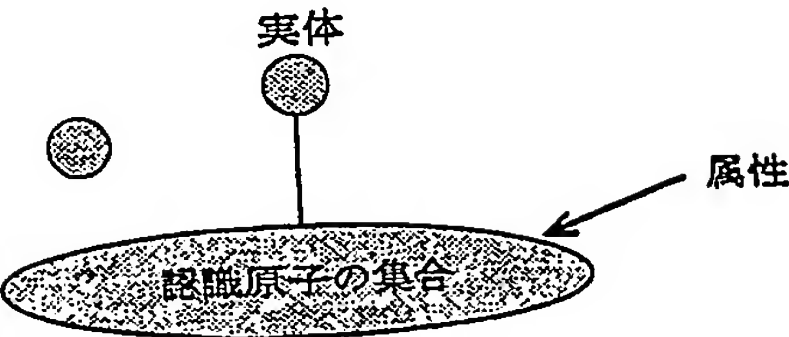
[図3]



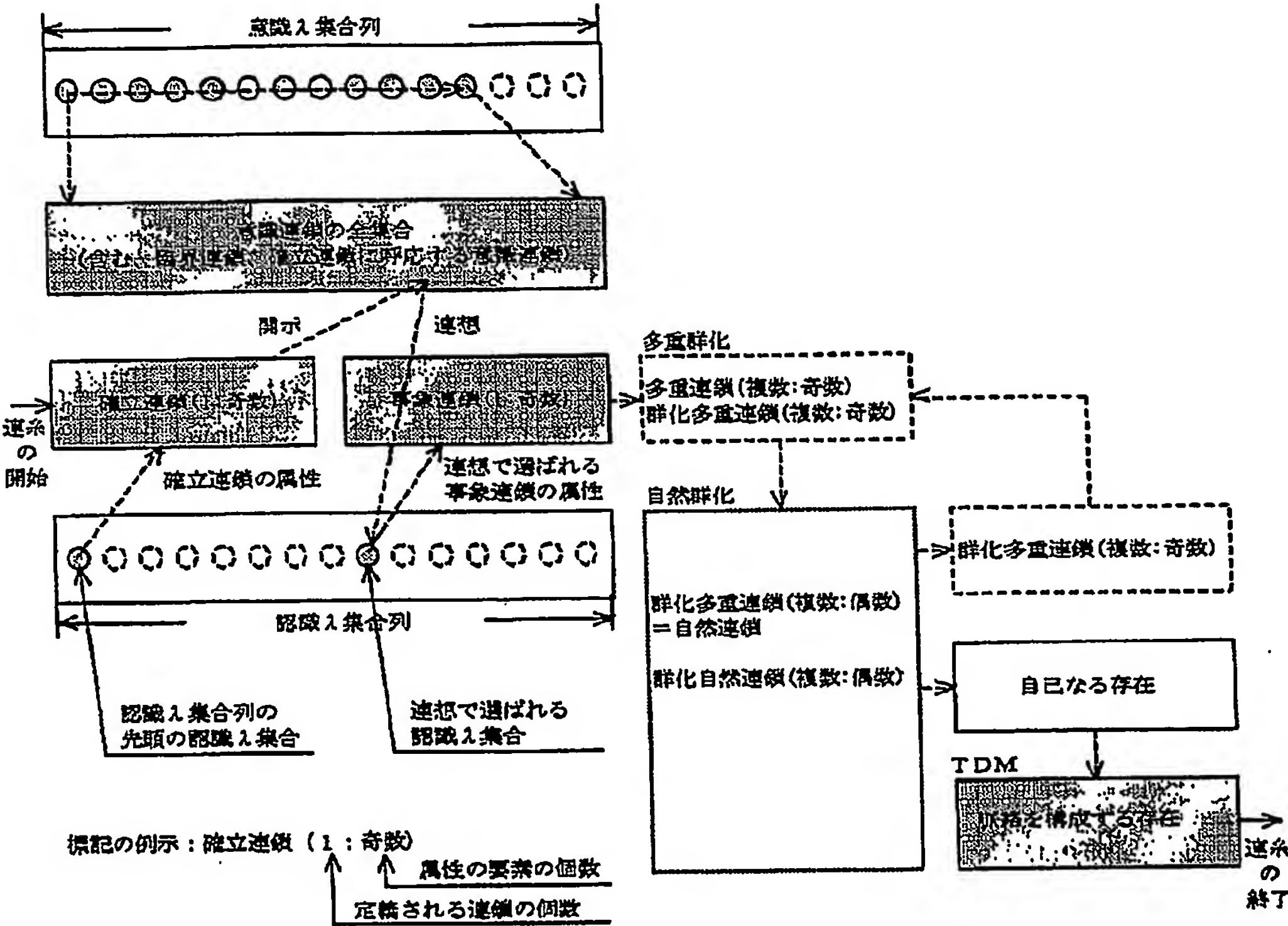
[図4]



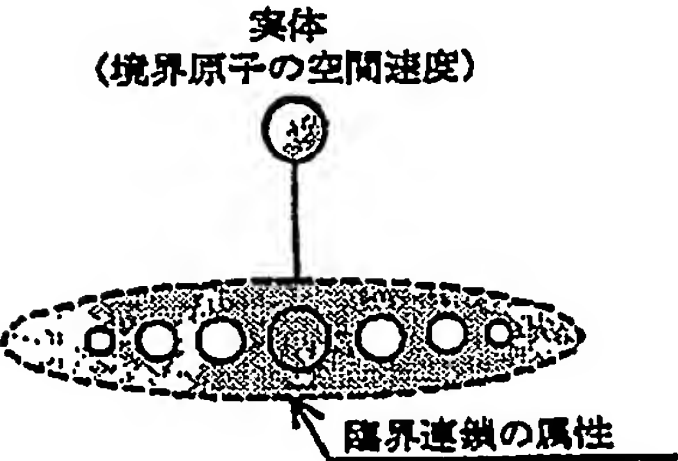
[図5]



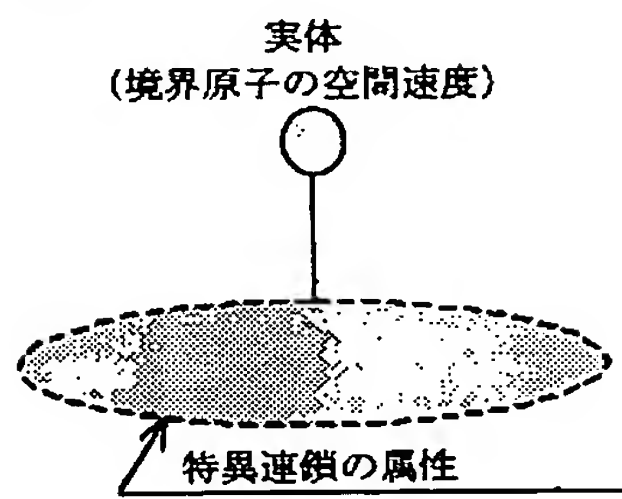
[図6]



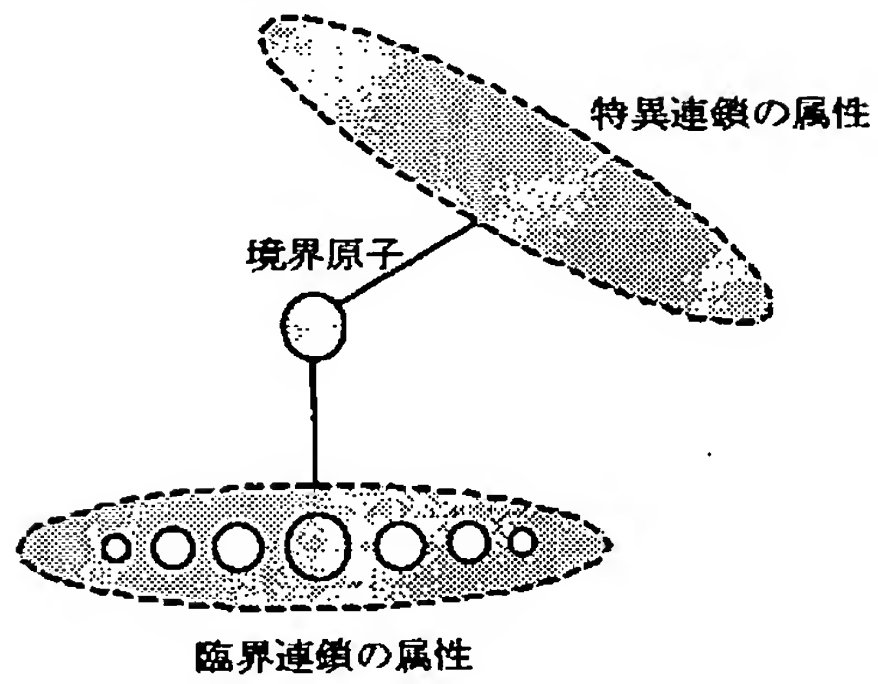
[図7]



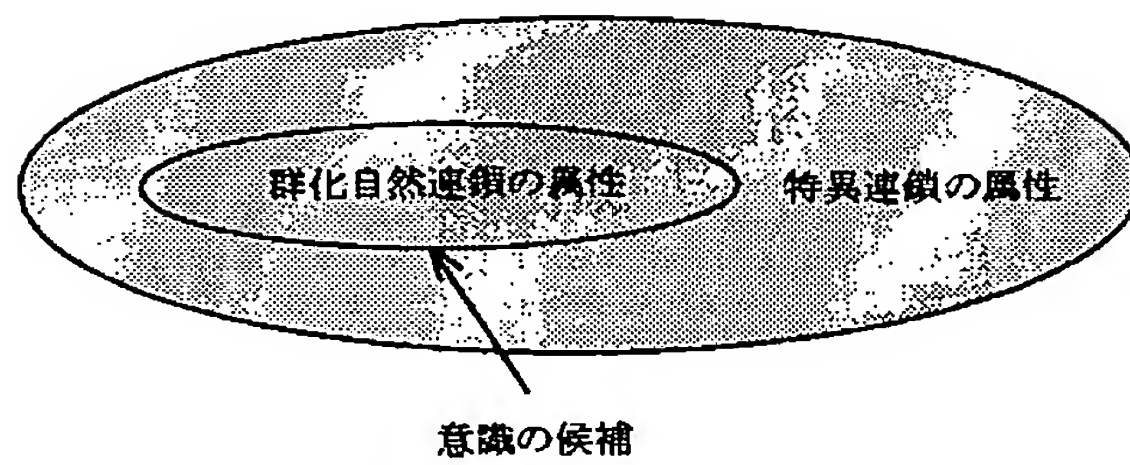
[図8]



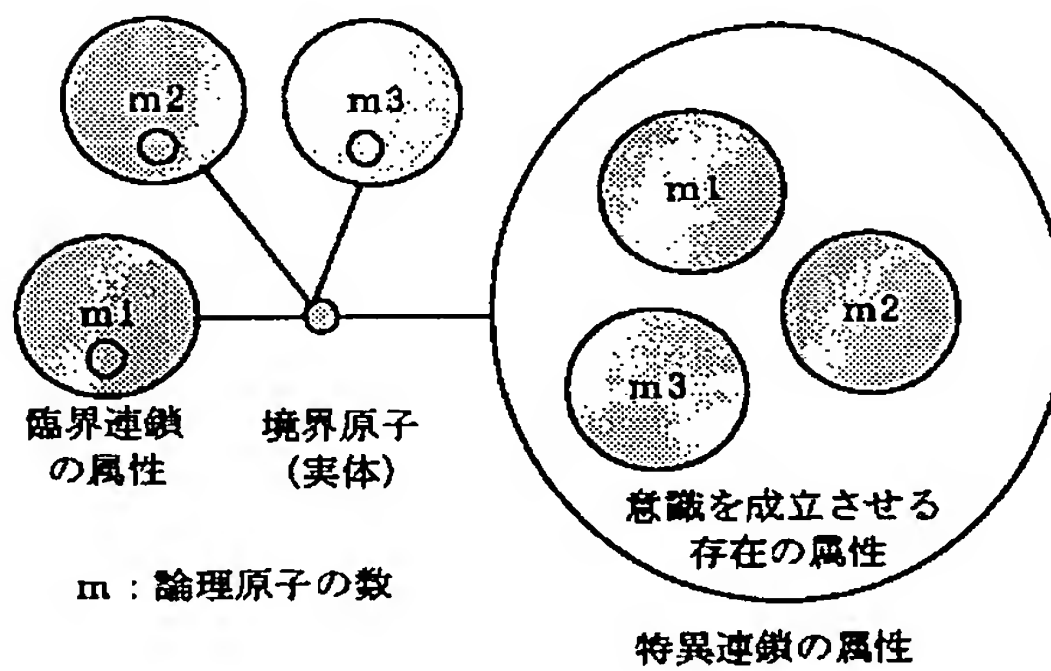
[図9]



[図10]

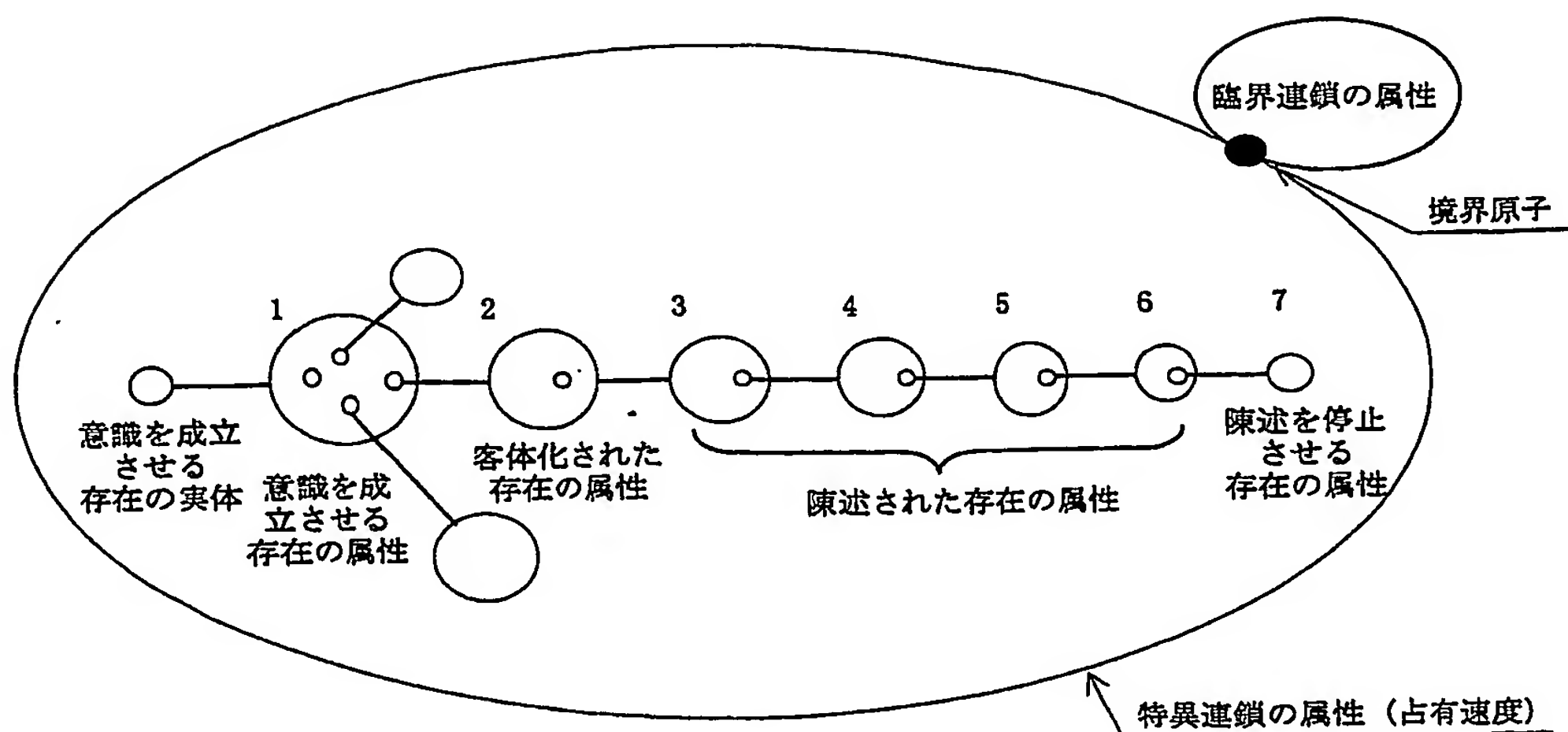


[図11]

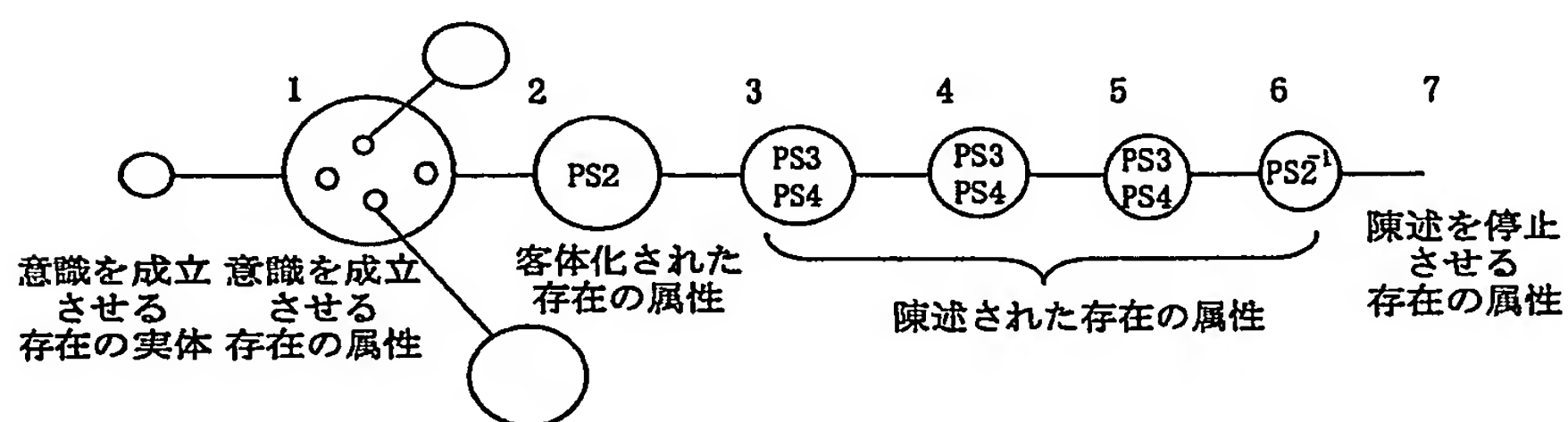




[図12]



[図13]



論考図1：今の自己の存在証明を過去の自己と未来の自己の存在証明を用いて行う

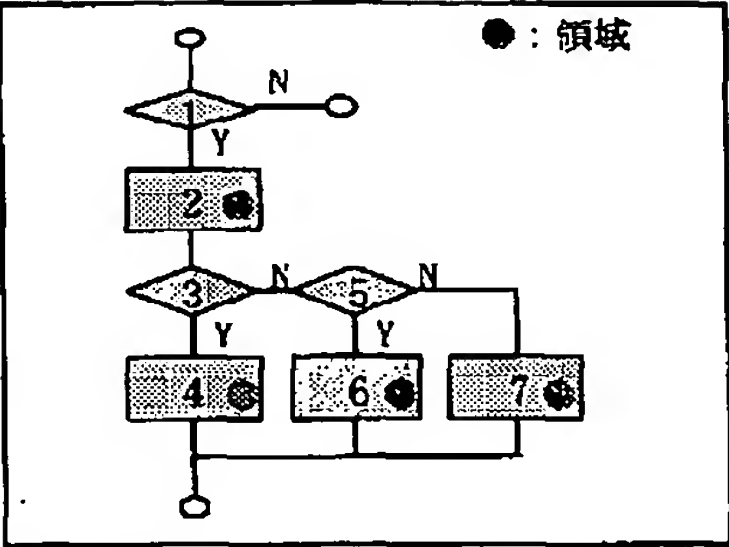
今の自己	2	3	4	5	6	7
過去の自己	1	2	3	4	5	
未来の自己	3	4	5	6		

論考図2：今の自己の存在はPSを用いて行われる

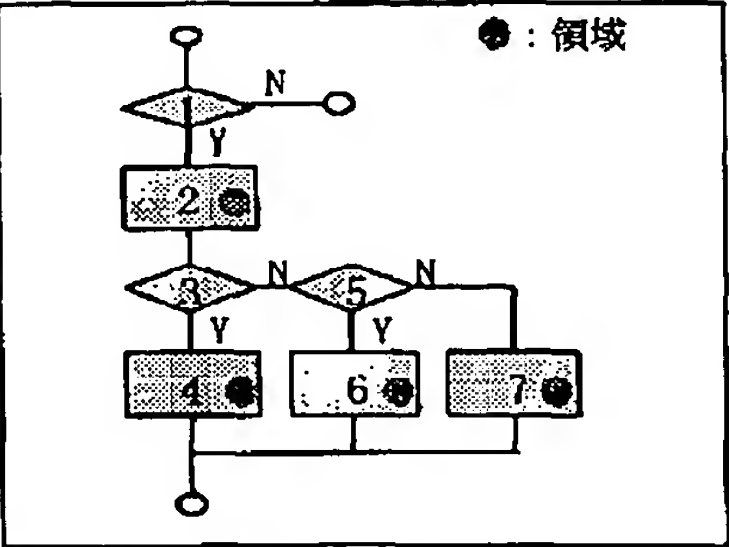
今の自己	2	3	4	5	6	7
過去の自己	PS2	PS3	PS3	PS3		
未来の自己		PS4	PS4	PS4	PS2 <sup>-1</sup>	

- ・ PS2 : 図 1 5 参照
- ・ PS3 : 図 1 6 参照
- ・ PS4 : 図 1 4 参照
- ・ PS2<sup>-1</sup> : 図 1 7 参照

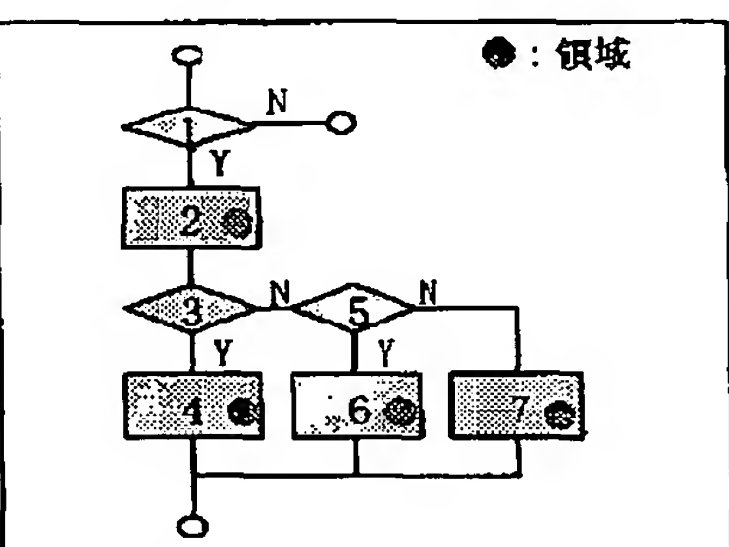
[図14]



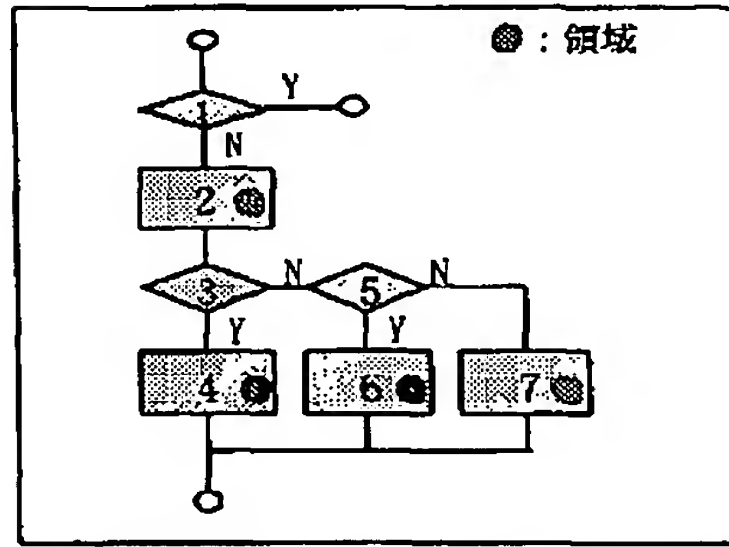
[図15]



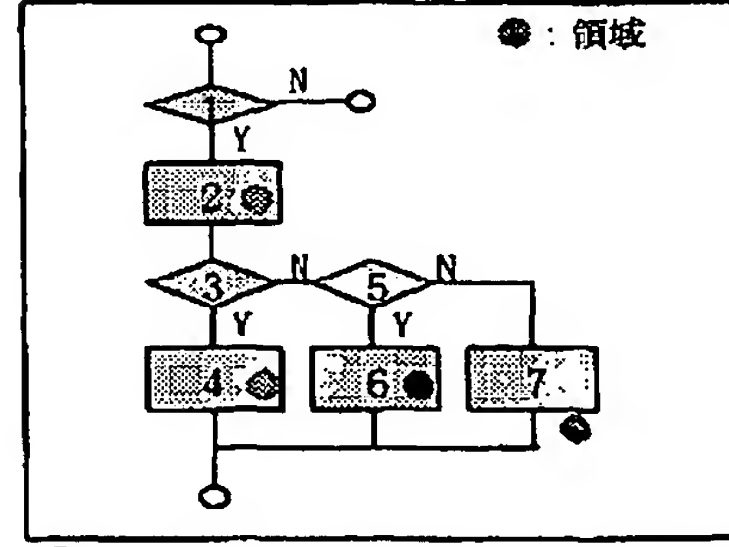
[図16]



[図17]



①PS4N: PS4を否定する言表構造



②PS2N: PS4Nに対応するPS2、PS2NはPS2の可逆構造である

NO		
1	第1規約	未来の自己の成立を判定する
2	第2規約	多重連鎖の叙述を用いて未来の自己を叙述する
3	第3規約	未来の自己の叙述の成立を判定する
4	第4規約	未来の自己の成立を宣言する
5	第5規約	全未来の自己の叙述の成立の状態変化を判定する
6	第6規約	未来の自己が叙述されていない事を宣言する
7	第7規約	過去の自己の叙述が成立していない事を宣言する

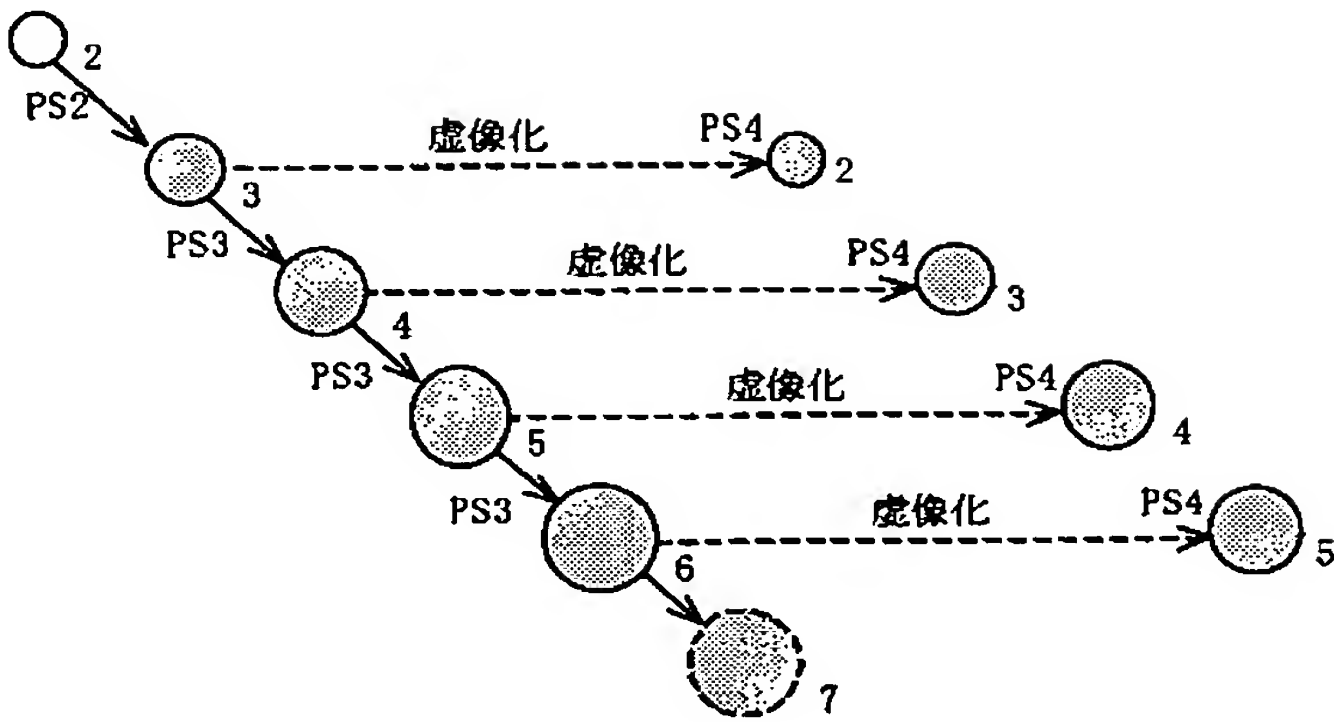
NO		
1	第1規約	今の自己の成立を判定する
2	第2規約	確立連鎖の叙述を用いて今の自己を叙述する
3	第3規約	今の自己の叙述の成立を判定する
4	第4規約	今の自己の成立を宣言する
5	第5規約	全今の自己の叙述の成立の状態変化を判定する
6	第6規約	今の自己が叙述されていない事を宣言する
7	第7規約	今の自己の叙述が成立していない事を宣言する

NO		
1	第1規約	過去の自己の成立を判定する
2	第2規約	事象連鎖の叙述を用いて過去の自己を叙述する
3	第3規約	過去の自己の叙述の成立を判定する
4	第4規約	過去の自己の成立を宣言する
5	第5規約	全過去の自己の叙述の成立の状態変化を判定する
6	第6規約	過去の自己が叙述されていない事を宣言する
7	第7規約	過去の自己の叙述が成立していない事を宣言する

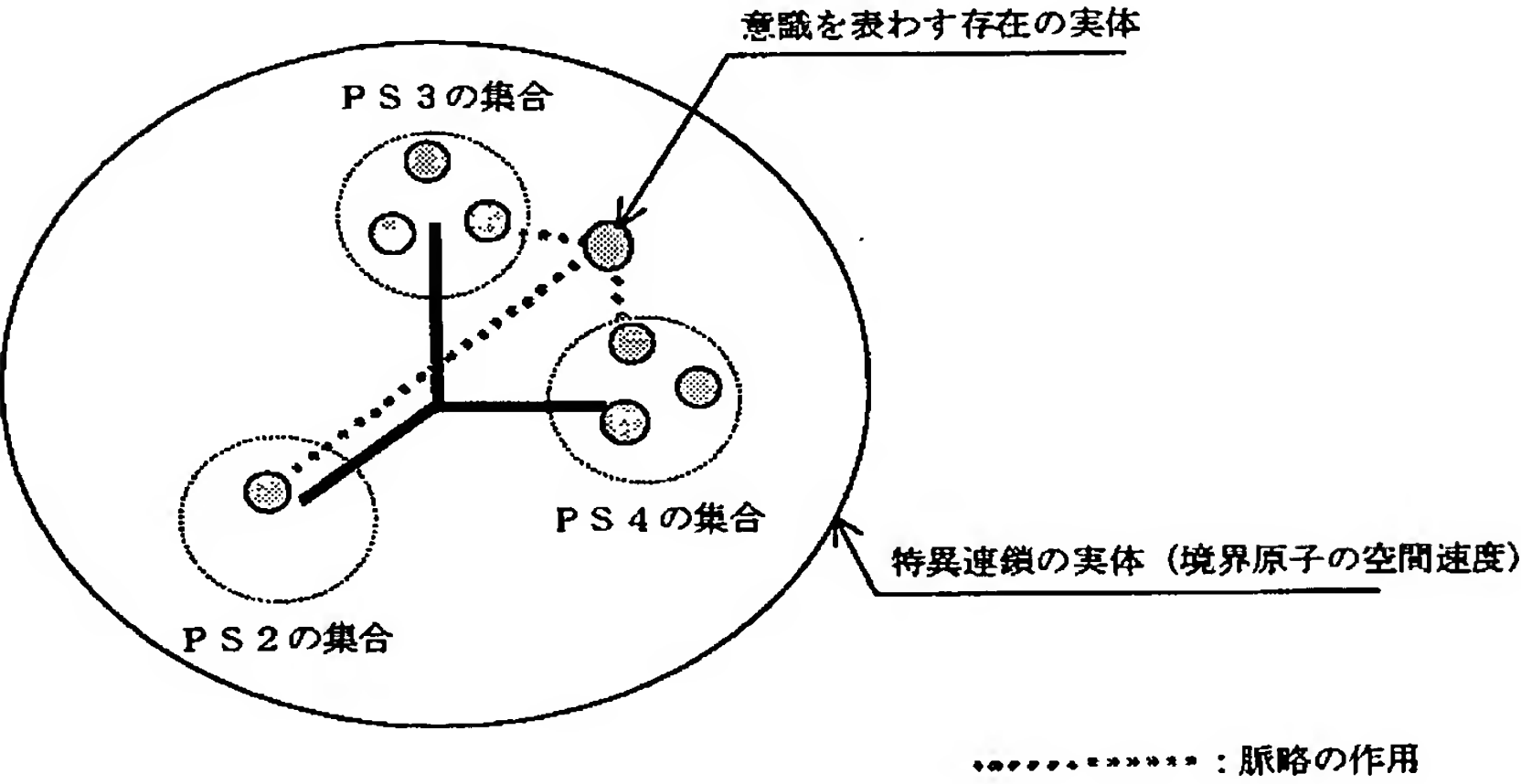
NO		
1	第1規約	未来の自己の成立を判定する
2	第2規約	事象連鎖(過去)の叙述を多重連鎖(未来)の叙述に置き換えて未来の自己を叙述する
3	第3規約	未来の自己の叙述の成立を判定する
4	第4規約	未来の自己の成立を宣言し、且つ、PS2Nの第4領域にこの宣言を移す
5	第5規約	全未来の自己の叙述の成立の状態変化を判定する
6	第6規約	未来の自己が叙述されていない事を宣言する
7	第7規約	過去の自己の叙述が成立していない事を宣言する

NO		
1	第1規約	無
2	第2規約	無
3	第3規約	無
4	第4規約	PS4Nの第4領域
5	第5規約	無
6	第6規約	無
7	第7規約	無

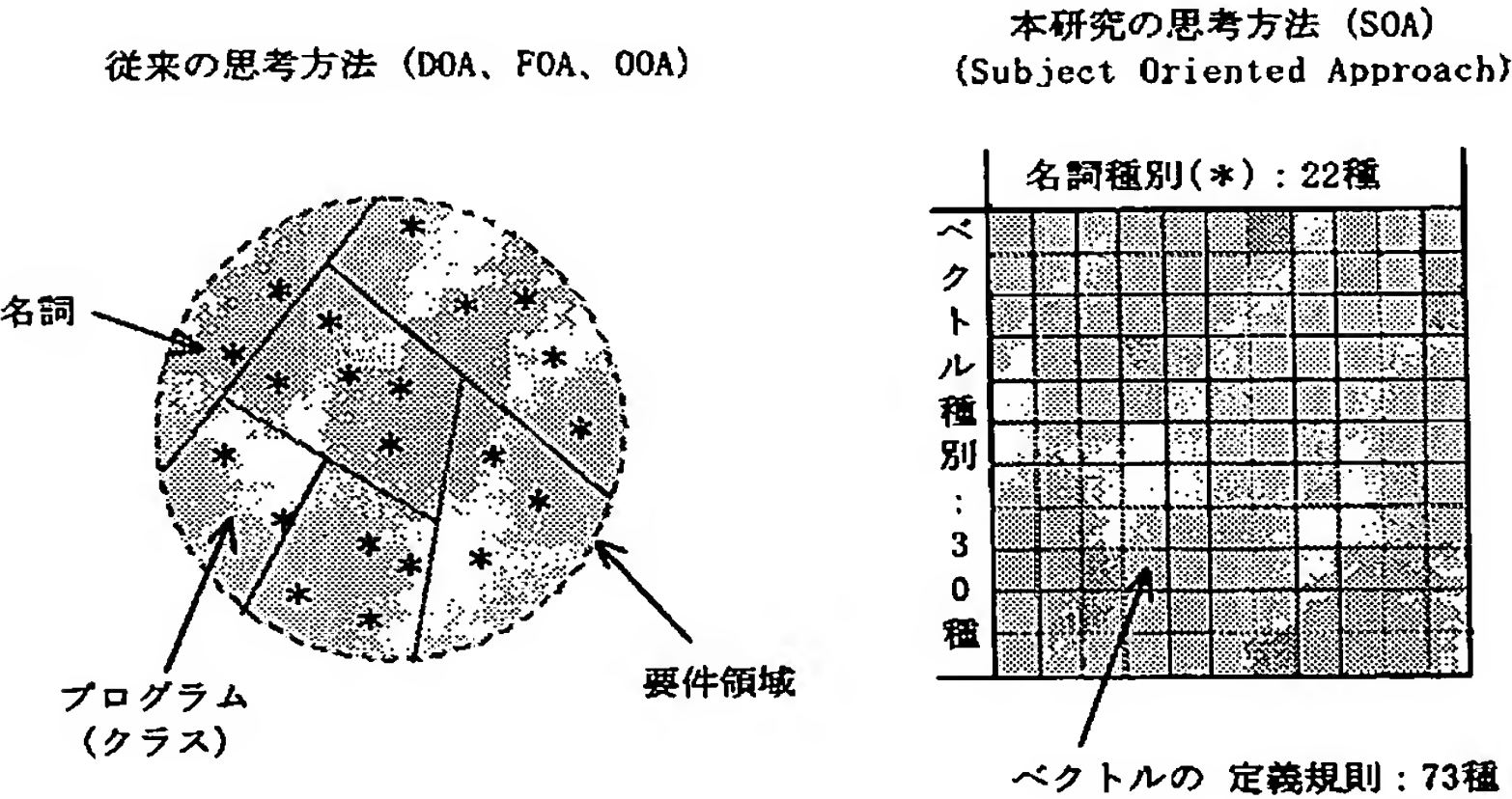
[図18]



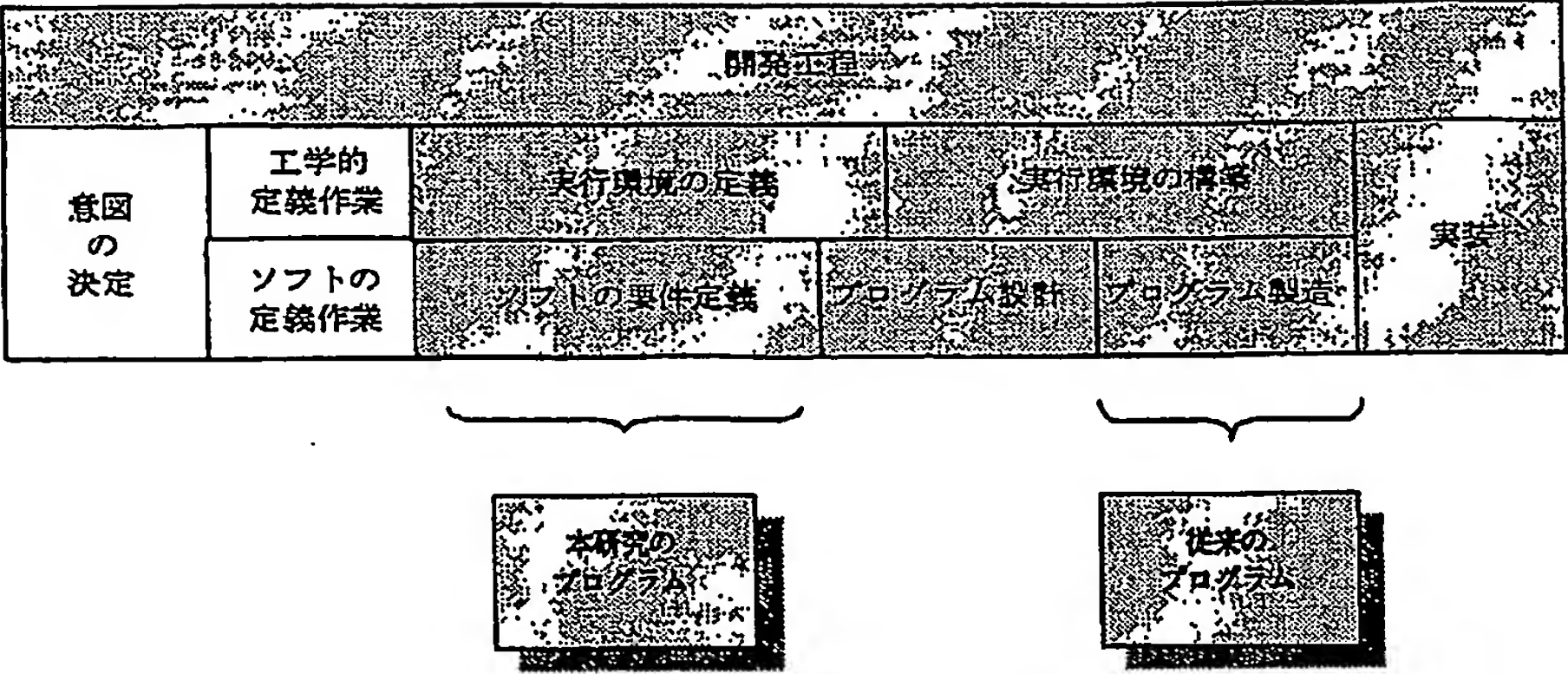
[図19]



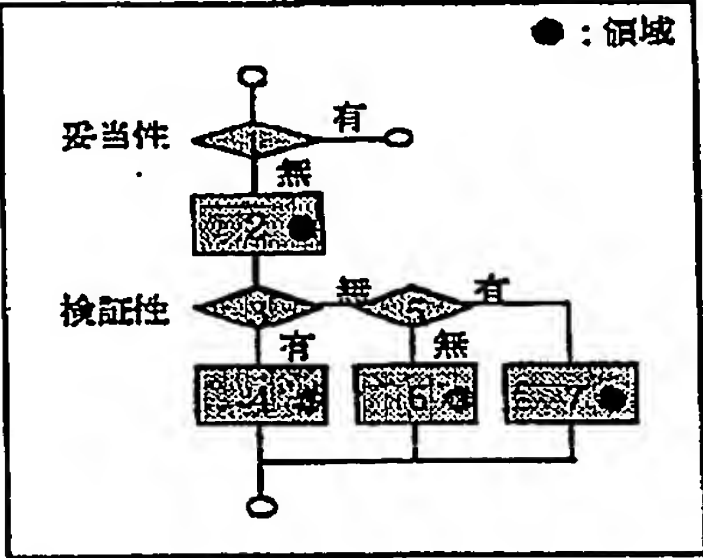
[図20]



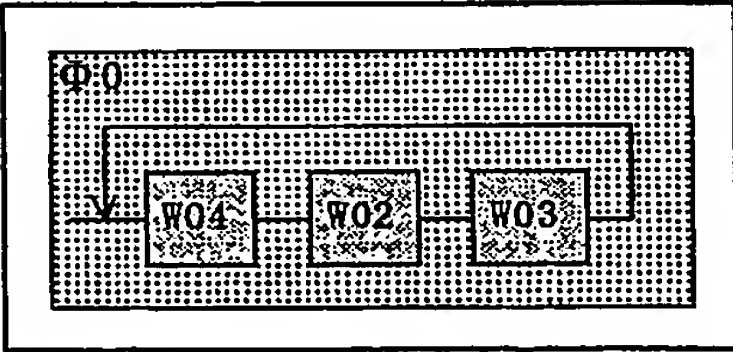
[図21]



[図22]



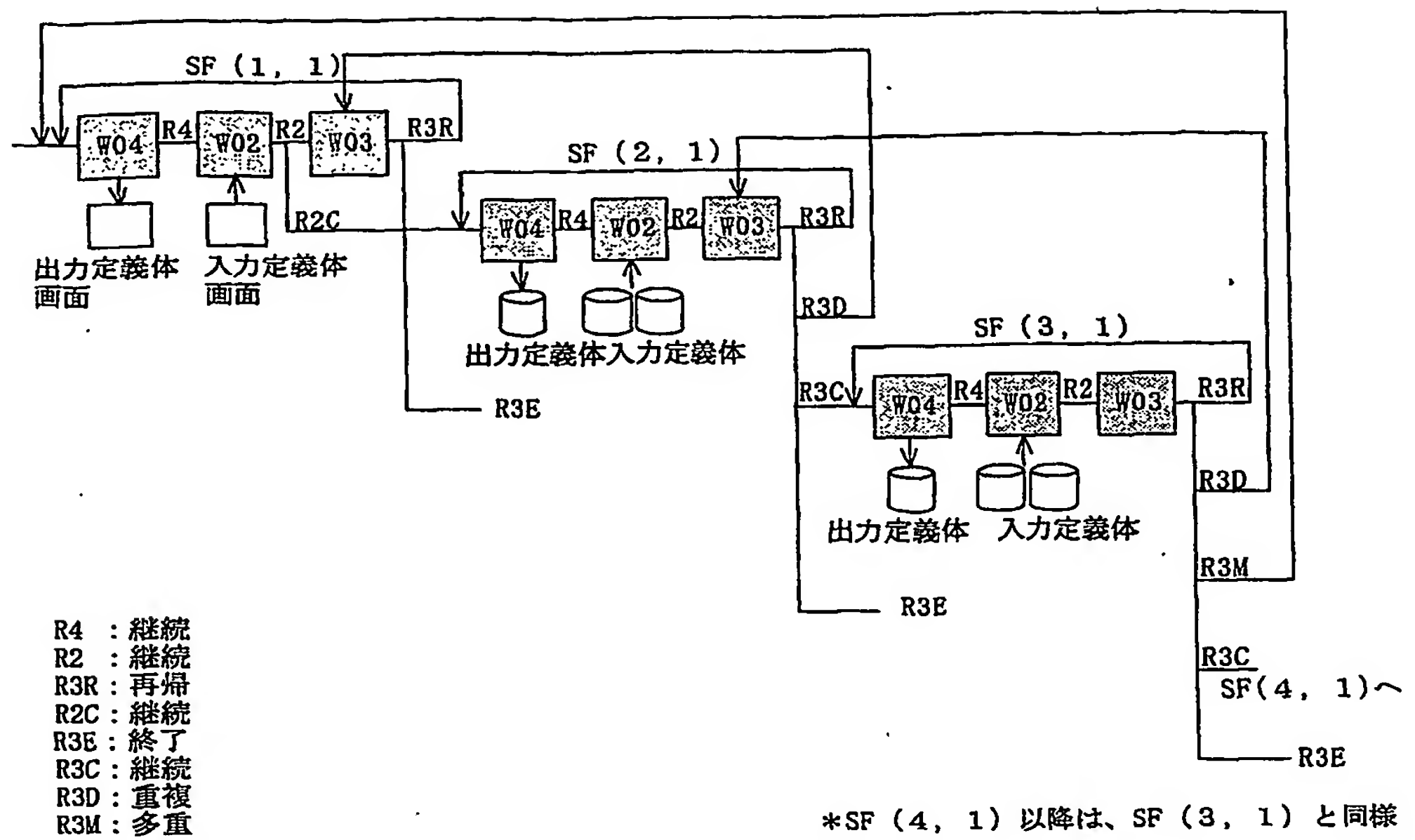
[図23]



NO	規則名	ベクトルを定義する標準の規則
1	第1規則	主語の第4領域の値の有無を確認する。
2	第2規則	主語の値を第2領域に生成する。
3	第3規則	主語の第2領域の値の有無を確認する。
4	第4規則	主語の第2領域の値を第4領域に移す。
5	第5規則	全主語の第4領域の周期上の状態変化を確認する。
6	第6規則	主語の不在を開示する。
7	第7規則	主語の探索の要請を行う。

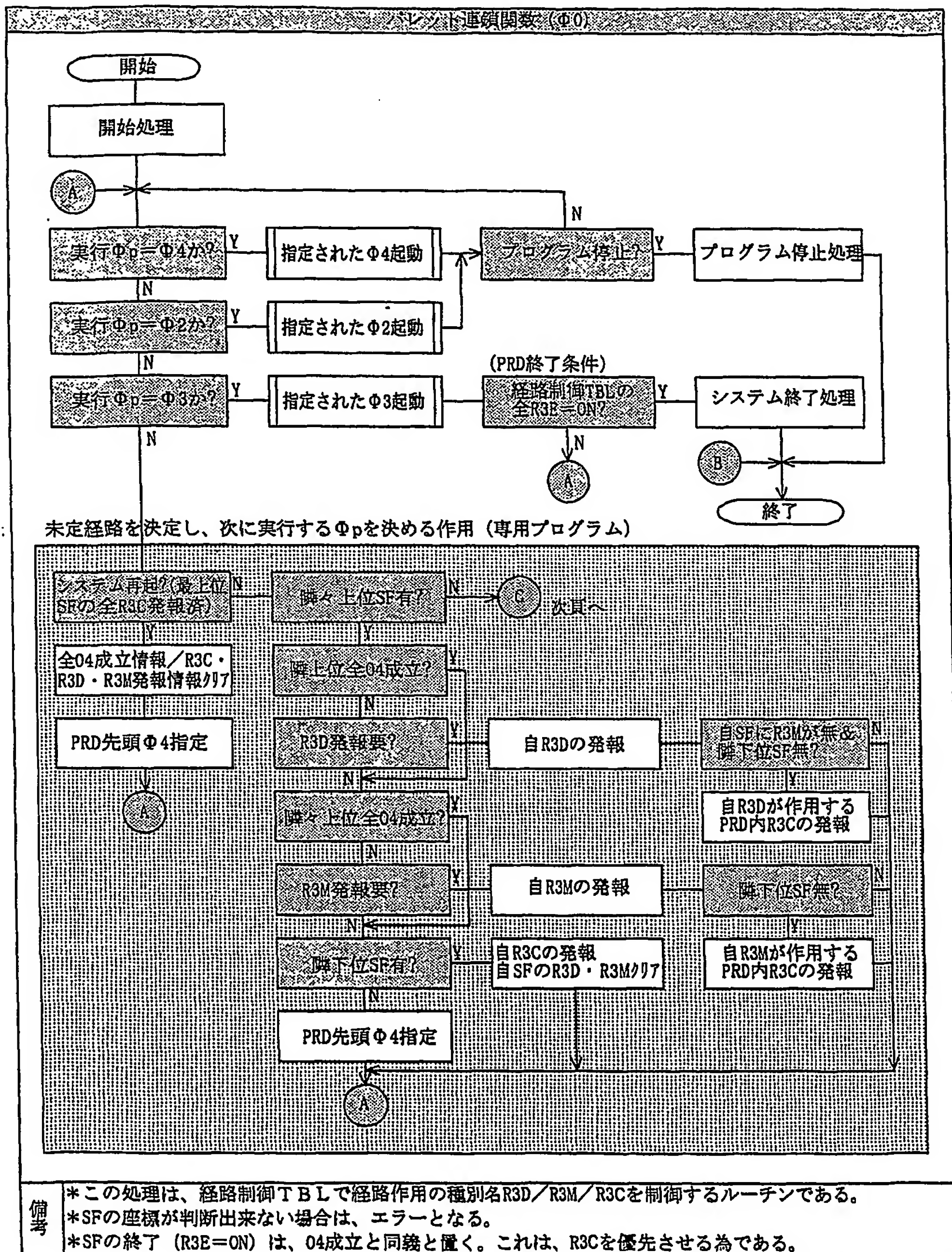
パレット連鎖関数	パレット	パレット関数	パレット関数に属すベクトル
Φ0	W04パレット	Φ4	L4、O4、S4、R4
	W02パレット	Φ2	L2、I2、R2
	W03パレット	Φ3	L3、R3R、R3E

[図24]

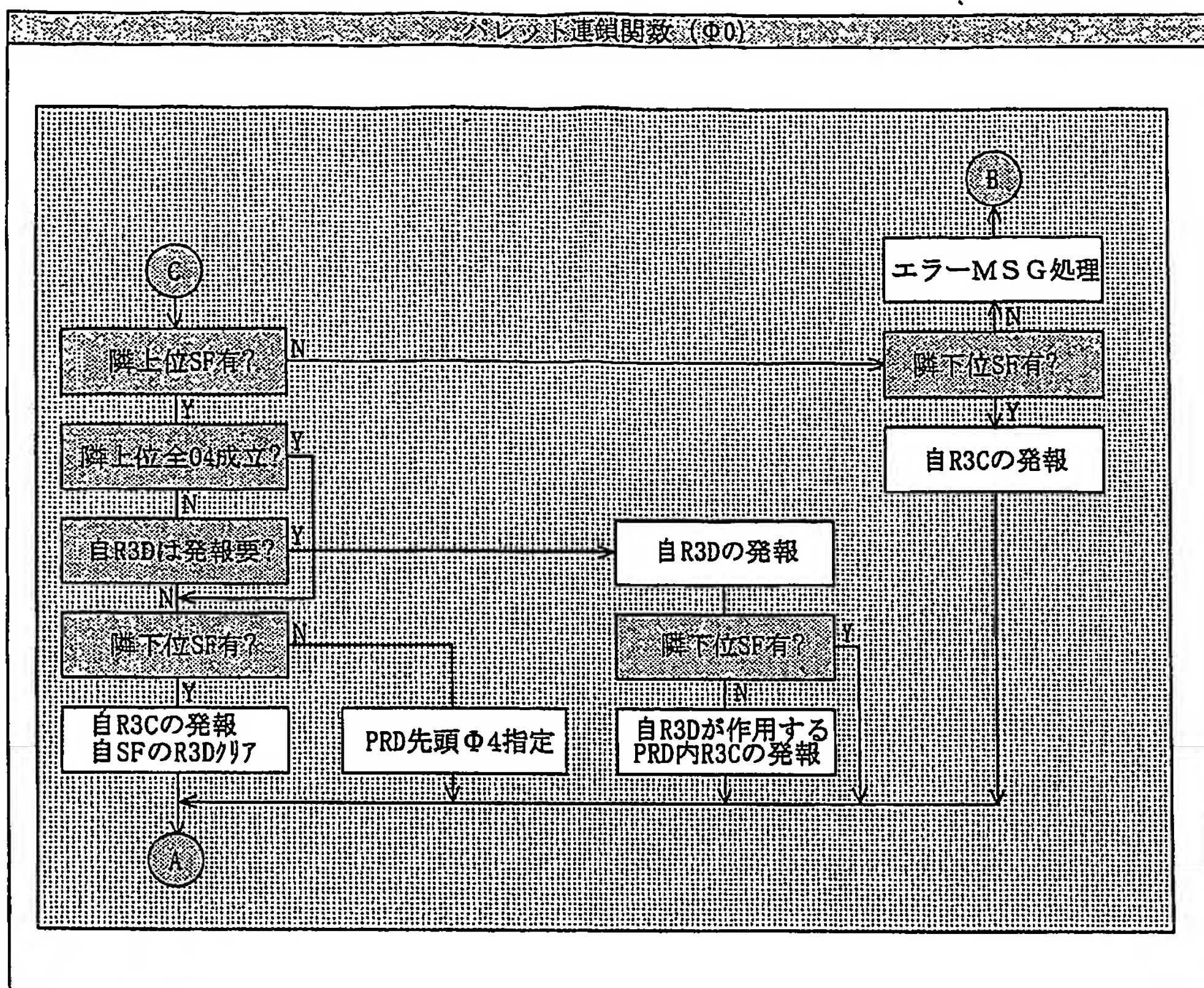




[図25]

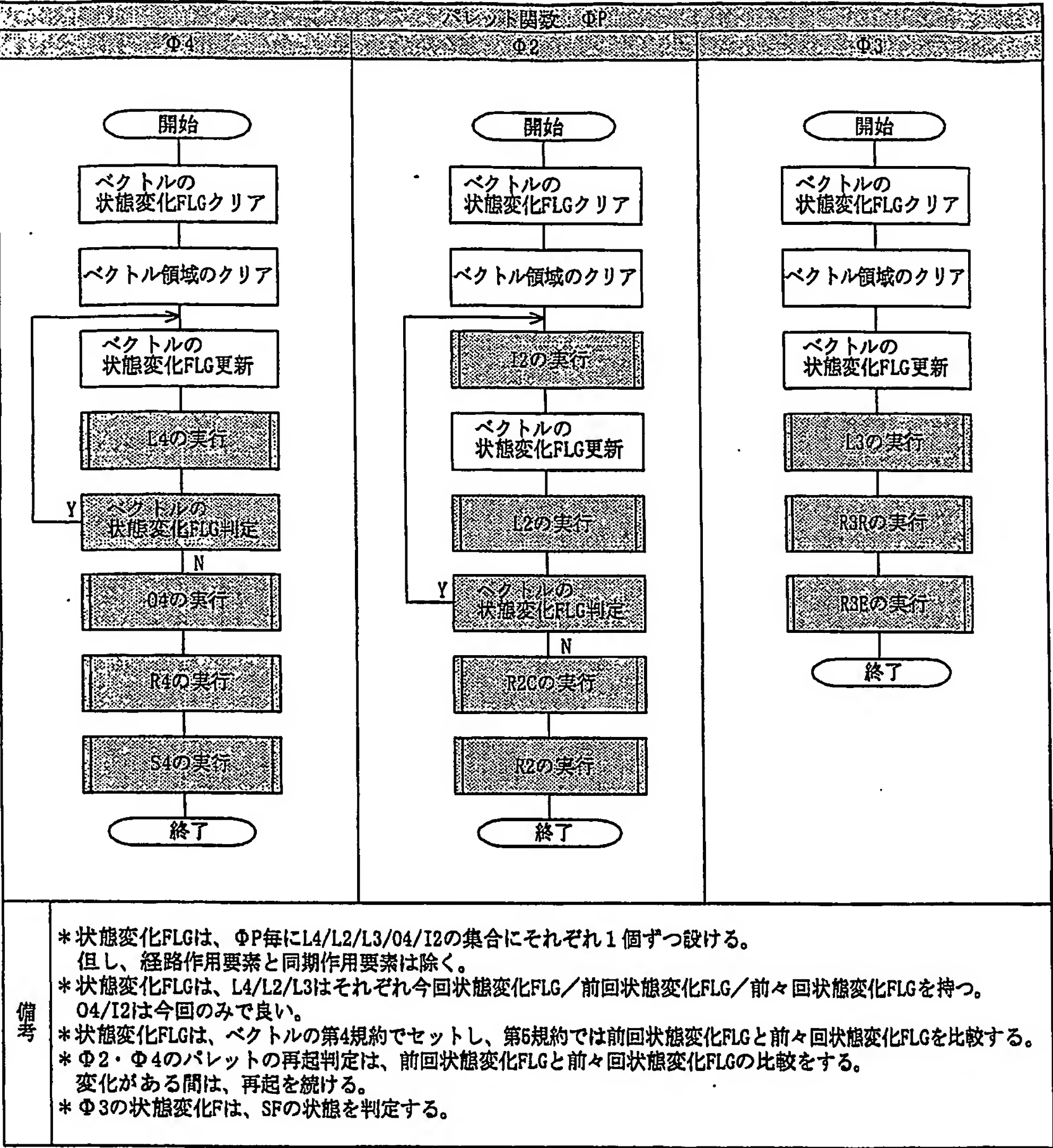


[図26]

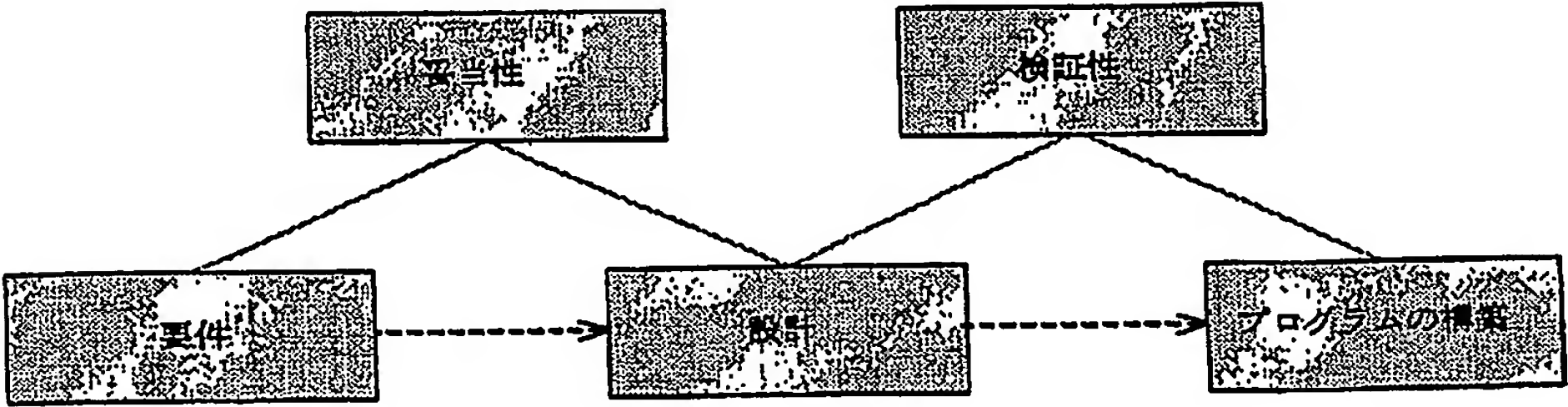




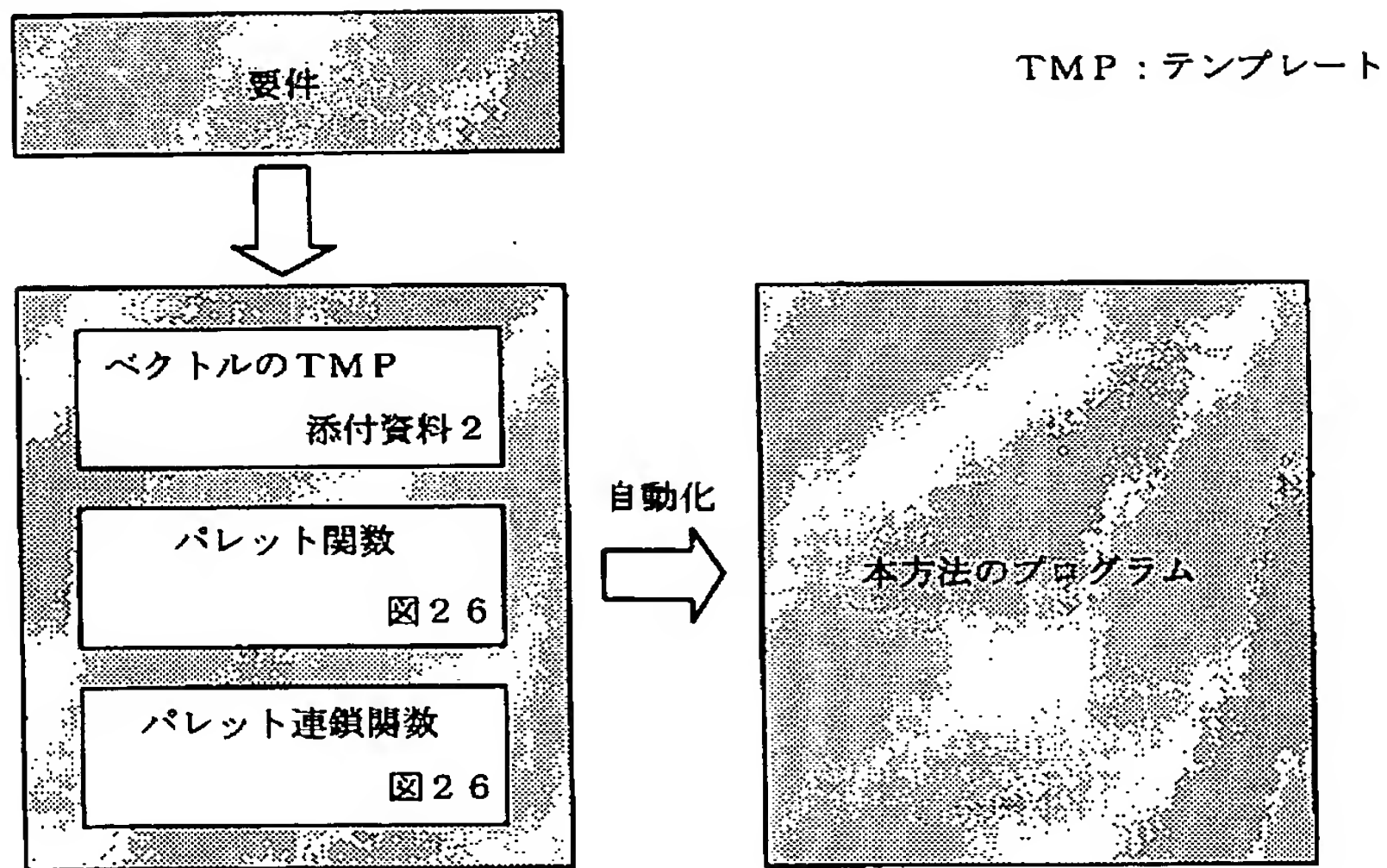
[図27]



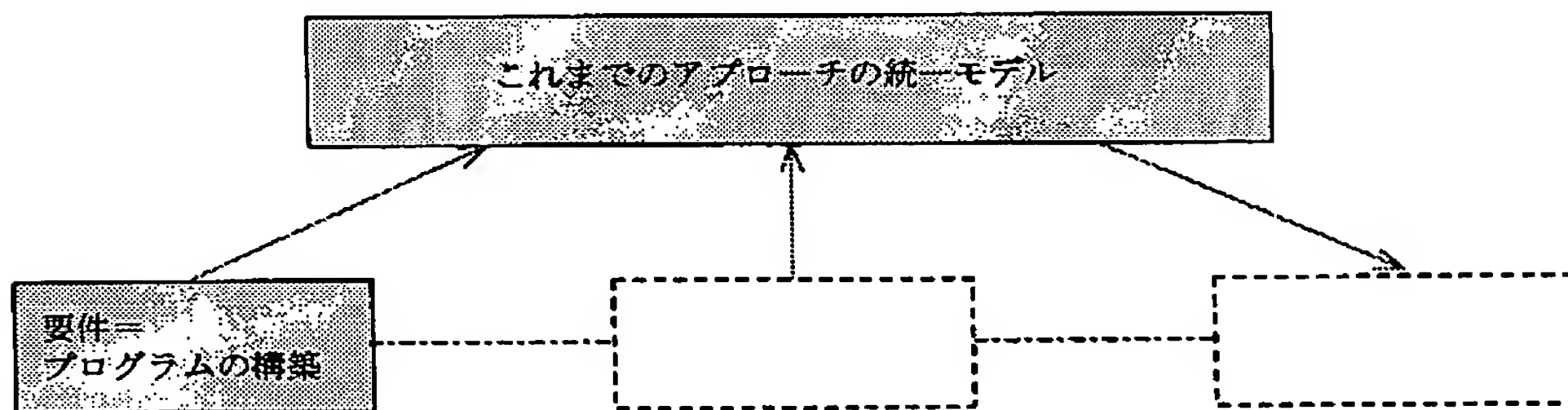
[図28]



[図29]



[図30]



[図31]

L2正規(入力)	L2規約
	第1 主語のL2第4領域 ≠ 空
	第2 入力論理体に属すデータ → 主語のL2第2領域
	第3 主語のL2第2領域 = 空
	第4 主語のL2第2領域 → 主語のL2第4領域 L2今回状態変化フラグ*に1を加算
	第5 L2前回状態変化フラグ* = L2前々回状態変化フラグ*
	第6 ON → 不成立フラグ*
	第7 ON → 再起フラグ*

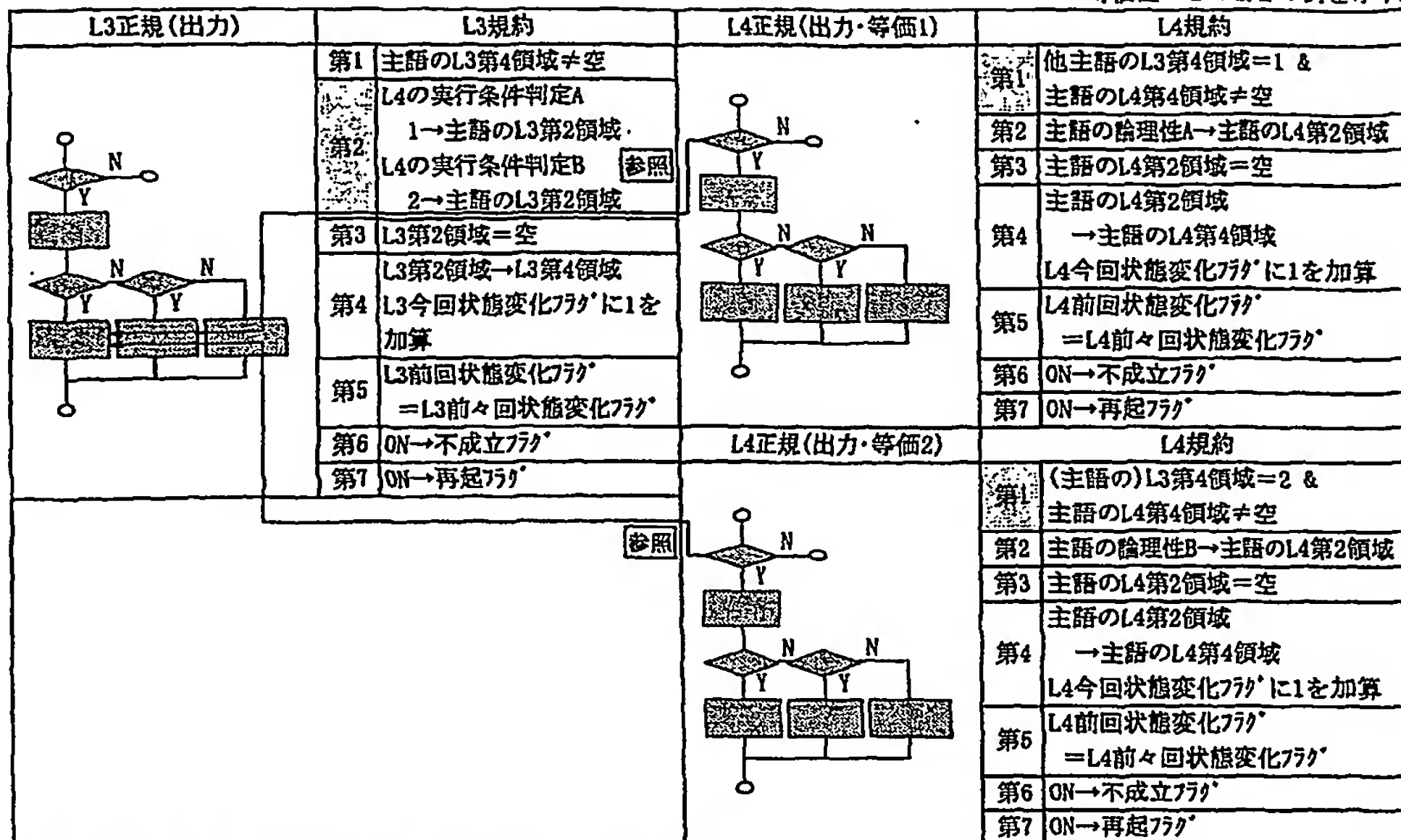
定義規則 1 : 主語が正規でその属性が入力のベクトル





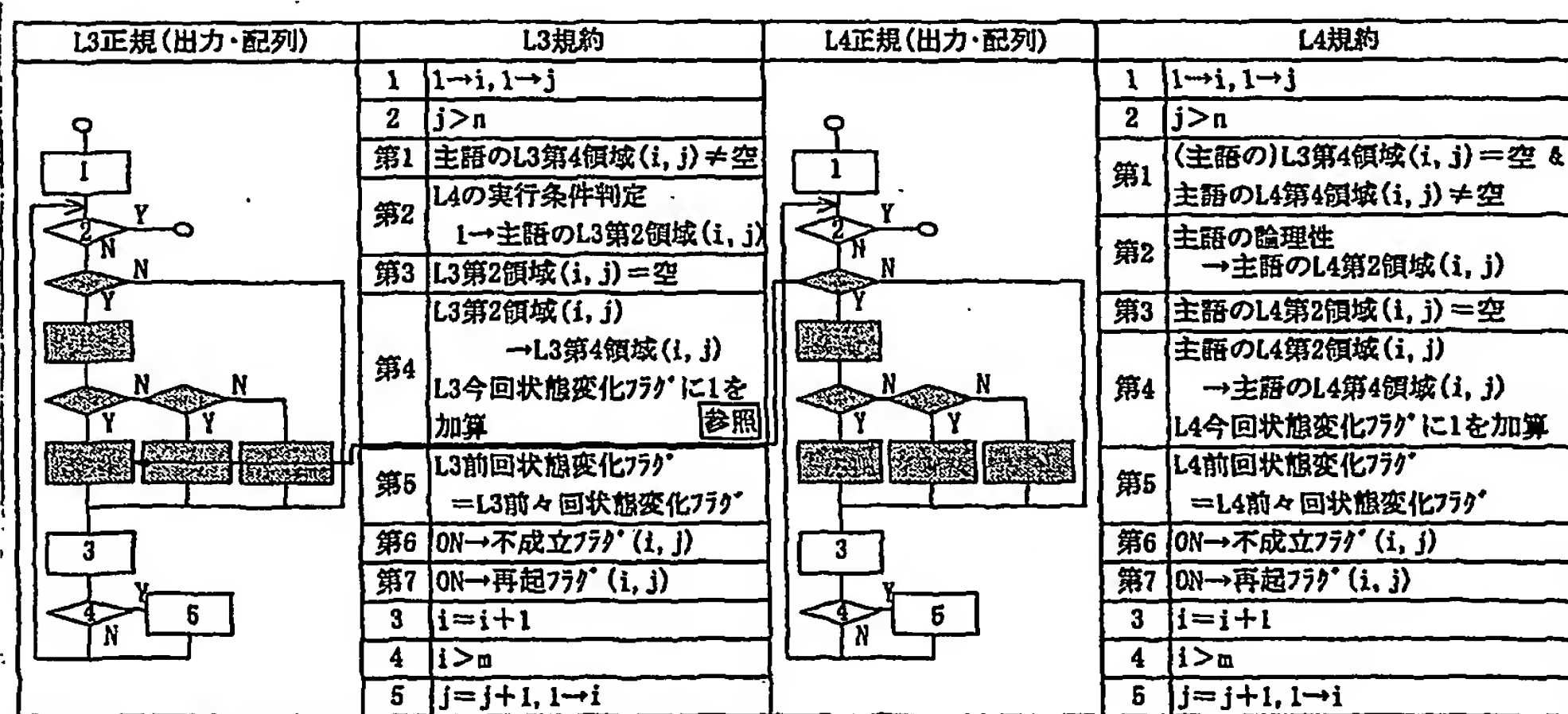
[図34]

\*等価性=2の場合の例を示す。



定義規則 4 : 主語が正規でその属性が出力, 等価のベクトル

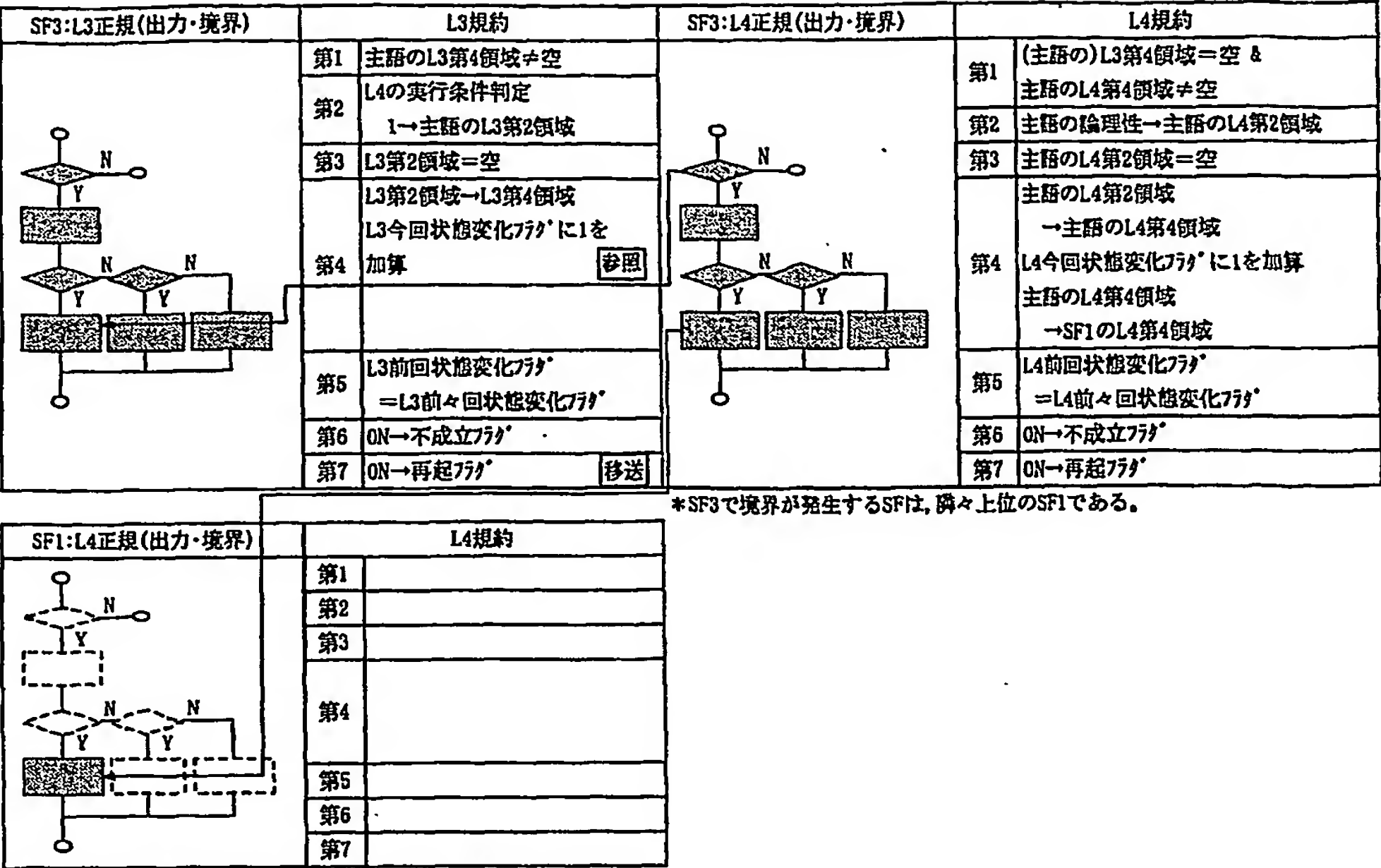
[図35]



定義規則 5 : 主語が正規でその属性が出力, 配列のベクトル



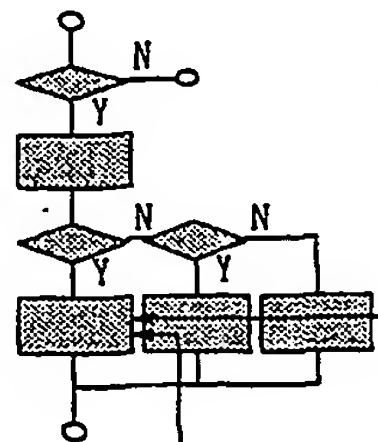
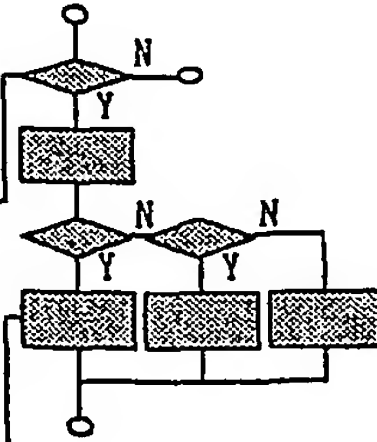
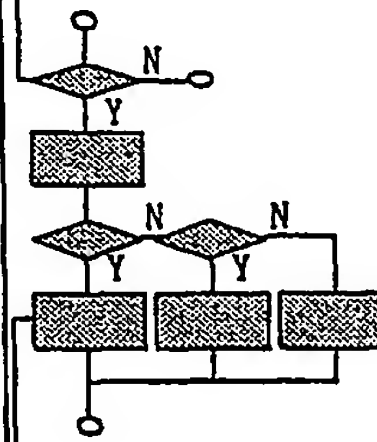
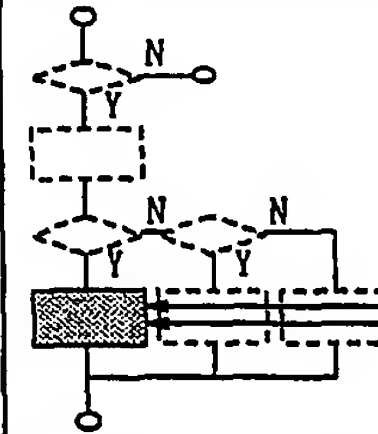
[図37]



定義規則7：主語が正規でその属性が出力、境界のベクトル

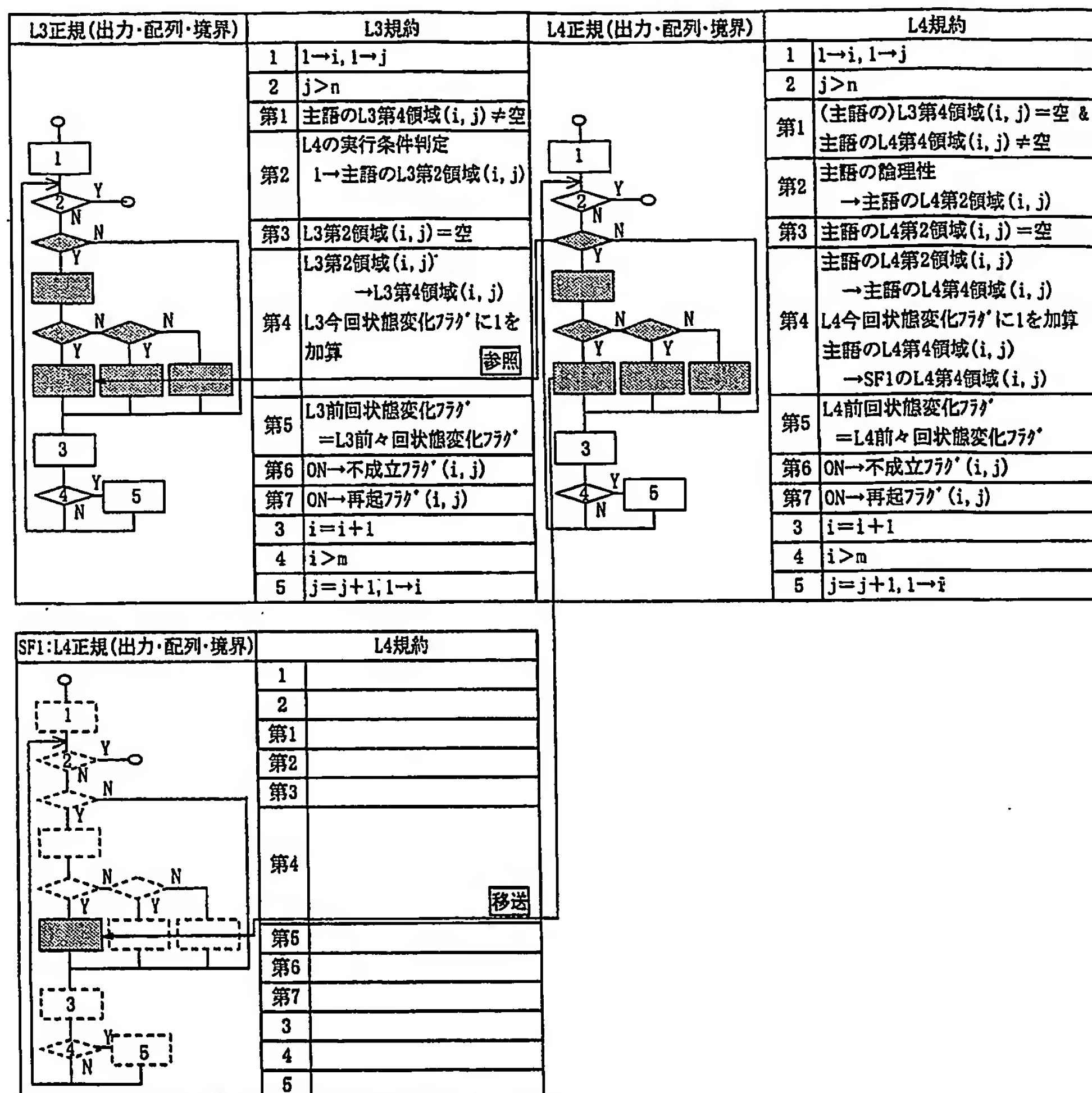
[図38]

\*等価性=2の場合の例を示す。

L3正規(出力・境界)		L3規約		L4正規(出力・等価1・境界)		L4規約	
	第1	主語のL3第4領域≠空			第1	(主語の)L3第4領域=1 & 主語のL4第4領域≠空	
	第2	L4の実行条件判定A 1→主語のL3第2領域 L4の実行条件判定B 2→主語のL3第2領域			第2	主語の論理性A→主語のL4第2領域	
	第3	L3第2領域=空 参照			第3	主語のL4第2領域=空	
	第4	L3第2領域→L3第4領域 L3今回状態変化フラグに1を加算			第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →SF1のL4第4領域	
	第5	L3前回状態変化フラグ =L3前々回状態変化フラグ			第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
	第6	ON→不成立フラグ			第6	ON→不成立フラグ	
	第7	ON→再起フラグ 移送			第7	ON→再起フラグ	
				L4正規(出力・等価2・境界)		L4規約	
		参照			第1	(主語の)L3第4領域=2 & 主語のL4第4領域≠空	
					第2	主語の論理性B→主語のL4第2領域	
					第3	主語のL4第2領域=空	
					第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →SF1のL4第4領域	
					第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
					第6	ON→不成立フラグ	
					第7	ON→再起フラグ	
				*SF3で境界が発生するSFは、偶々上位のSF1である。			
SF1:L4正規(出力・境界)		L4規約					
	第1						
	第2						
	第3						
	第4						
	第5						
	第6						
	第7						

**定義規則 8 :** 主語が正規でその属性が出力，等価，境界のベクトル

[図 39]

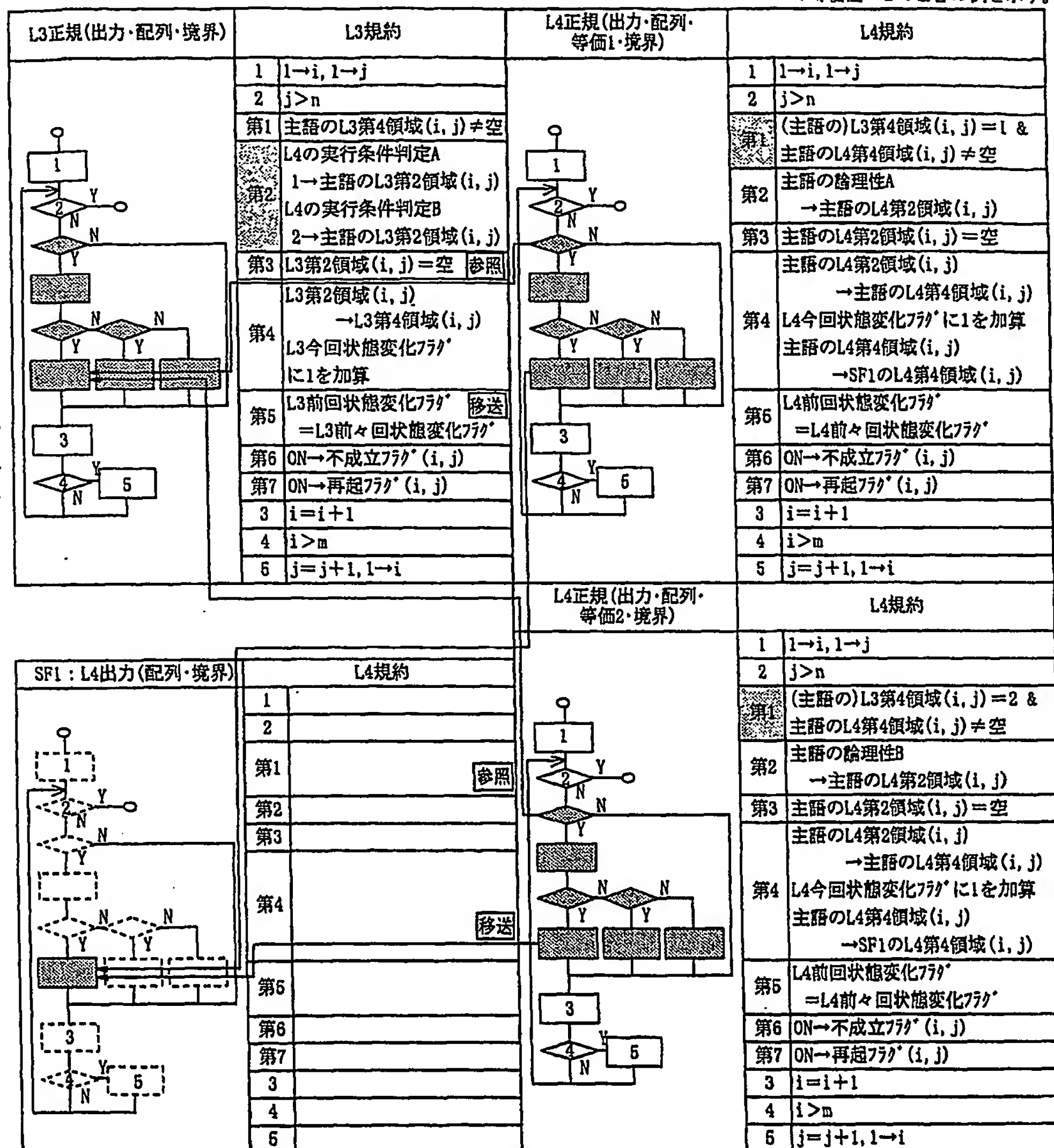


定義規則 9 : 主語が正規でその属性が出力, 配列, 境界のベクトル



[図40]

\*等価性=2の場合の例を示す。



定義規則10 : 主語が正規でその属性が出力, 等価, 配列, 境界のベクトル

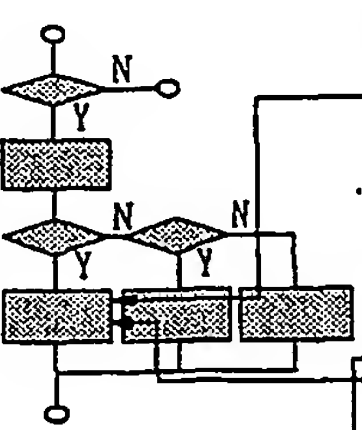
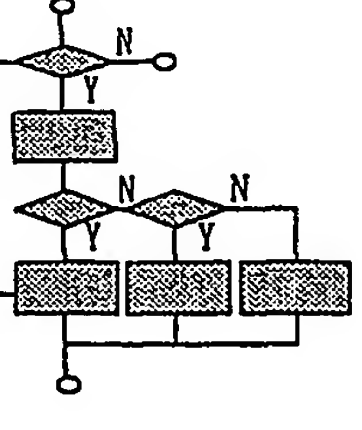
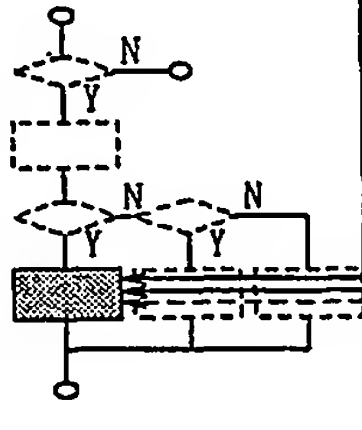
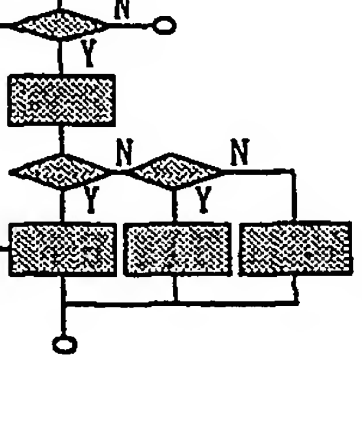
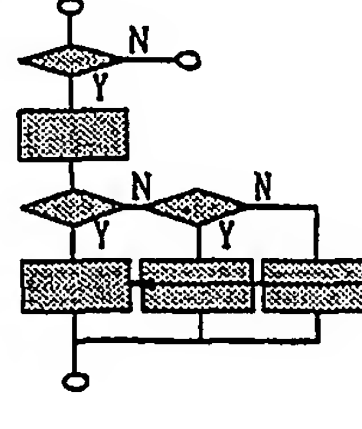
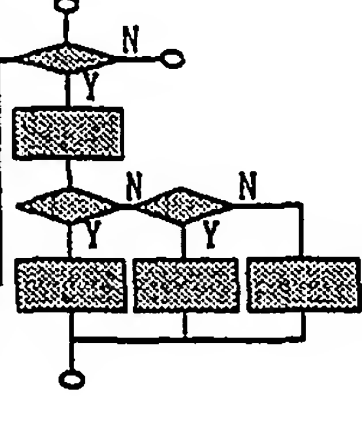
[図41]

L3K		L3規約		L4K		L4規約	
	第1	主語のL3第4領域≠空			第1	(主語の)L3第4領域=空 & 主語のL4第4領域≠空	
	第2	派生元の実行条件判定 1→主語のL3第2領域			第2	派生元の陰理性 →主語のL4第2領域	
	第3	L3第2領域=空	参照		第3	主語のL4第2領域=空	
	第4	L3第2領域→L3第4領域 L3今回状態変化フラグに1を加算			第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域	
	第5	L3前回状態変化フラグ =L3前々回状態変化フラグ			第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
	第6	ON→不成立フラグ			第6	ON→不成立フラグ	
	第7	ON→再起フラグ			第7	ON→再起フラグ	
L2K		L2規約					
	第1						
	第2						
	第3						
	第4						
	第5						
	第6						
	第7						
・Kの派生元となった正規単語のベクトル規約							
L3正規(Kの元)(出力)		L3規約		L4正規(Kの元)(出力)		L4規約	
	第1	主語のL3第4領域≠空			第1	(主語の)L3第4領域=空 & 主語のL4第2領域≠空	
	第2	(無条件) 1→主語のL3第2領域			第2	KのL2第4領域 →主語のL4第2領域	
	第3	L3第2領域=空			第3	主語のL4第2領域=空	
	第4	L3第2領域→L3第4領域 L3今回状態変化フラグに1を加算			第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算	
	第5	L3前回状態変化フラグ =L3前々回状態変化フラグ			第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
	第6	ON→不成立フラグ			第6	ON→不成立フラグ	
	第7	ON→再起フラグ			第7	ON→再起フラグ	

**定義規則 1.1 :** 主語が  $K$  のベクトル

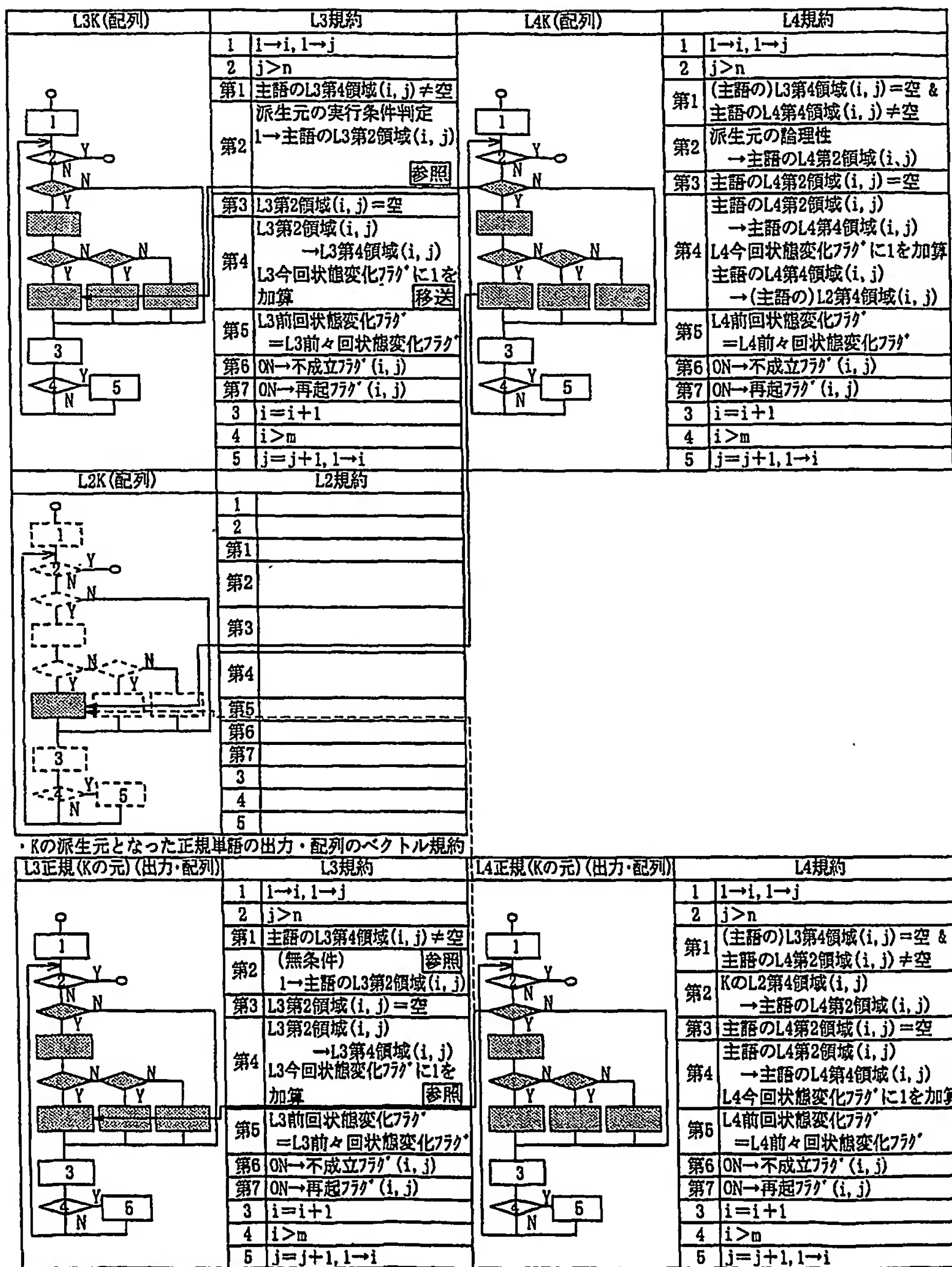
[図42]

\*等価性=2の場合の例を示す。

<div>L3K</div> <div></div>	<div>L3規約</div> <div><div>第1 主語のL3第4領域≠空</div><div>第2 派生元の実行条件判定A 1→主語のL3第2領域 派生元の実行条件判定B 2→主語のL3第2領域</div><div>第3 L3第2領域=空</div><div>第4 L3第2領域→L3第4領域 L3今回状態変化フラグに1を加算</div><div>第5 L3前回状態変化フラグ =L3前々回状態変化フラグ</div><div>第6 ON→不成立フラグ</div><div>第7 ON→再起フラグ</div></div>	<div>L4K(等価1)</div> <div></div>	<div>L4規約</div> <div><div>第1 (主語の)L3第4領域=1 &amp; 主語のL4第4領域≠空</div><div>第2 派生元の論理性A →主語のL4第2領域</div><div>第3 主語のL4第2領域=空</div><div>第4 主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域</div><div>第5 L4前回状態変化フラグ =L4前々回状態変化フラグ</div><div>第6 ON→不成立フラグ</div><div>第7 ON→再起フラグ</div></div>
<div>L2K</div> <div></div>	<div>L2規約</div> <div><div>第1</div><div>第2</div><div>第3</div><div>第4</div><div>第5</div><div>第6</div><div>第7</div></div>	<div>L4K(等価2)</div> <div></div>	<div>L4規約</div> <div><div>第1 (主語の)L3第4領域=2 &amp; 主語のL4第4領域≠空</div><div>第2 派生元の論理性B →主語のL4第2領域</div><div>第3 主語のL4第2領域=空</div><div>第4 主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域</div><div>第5 L4前回状態変化フラグ =L4前々回状態変化フラグ</div><div>第6 ON→不成立フラグ</div><div>第7 ON→再起フラグ</div></div>
・Kの派生元となった正規単語の出力・等価のベクトル規約			
<div>L3正規(Kの元)(出力)</div> <div></div>	<div>L3規約</div> <div><div>第1 主語のL3第4領域≠空</div><div>第2 (無条件) 1→主語のL3第2領域</div><div>第3 L3第2領域=空</div><div>第4 L3第2領域→L3第4領域 L3今回状態変化フラグに1を加算</div><div>第5 L3前回状態変化フラグ =L3前々回状態変化フラグ</div><div>第6 ON→不成立フラグ</div><div>第7 ON→再起フラグ</div></div>	<div>L4正規(Kの元)(出力)</div> <div></div>	<div>L4規約</div> <div><div>第1 (主語の)L3第4領域=空 &amp; 主語のL4第2領域≠空</div><div>第2 KのL2第4領域 →主語のL4第2領域</div><div>第3 主語のL4第2領域=空</div><div>第4 主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算</div><div>第5 L4前回状態変化フラグ =L4前々回状態変化フラグ</div><div>第6 ON→不成立フラグ</div><div>第7 ON→再起フラグ</div></div>

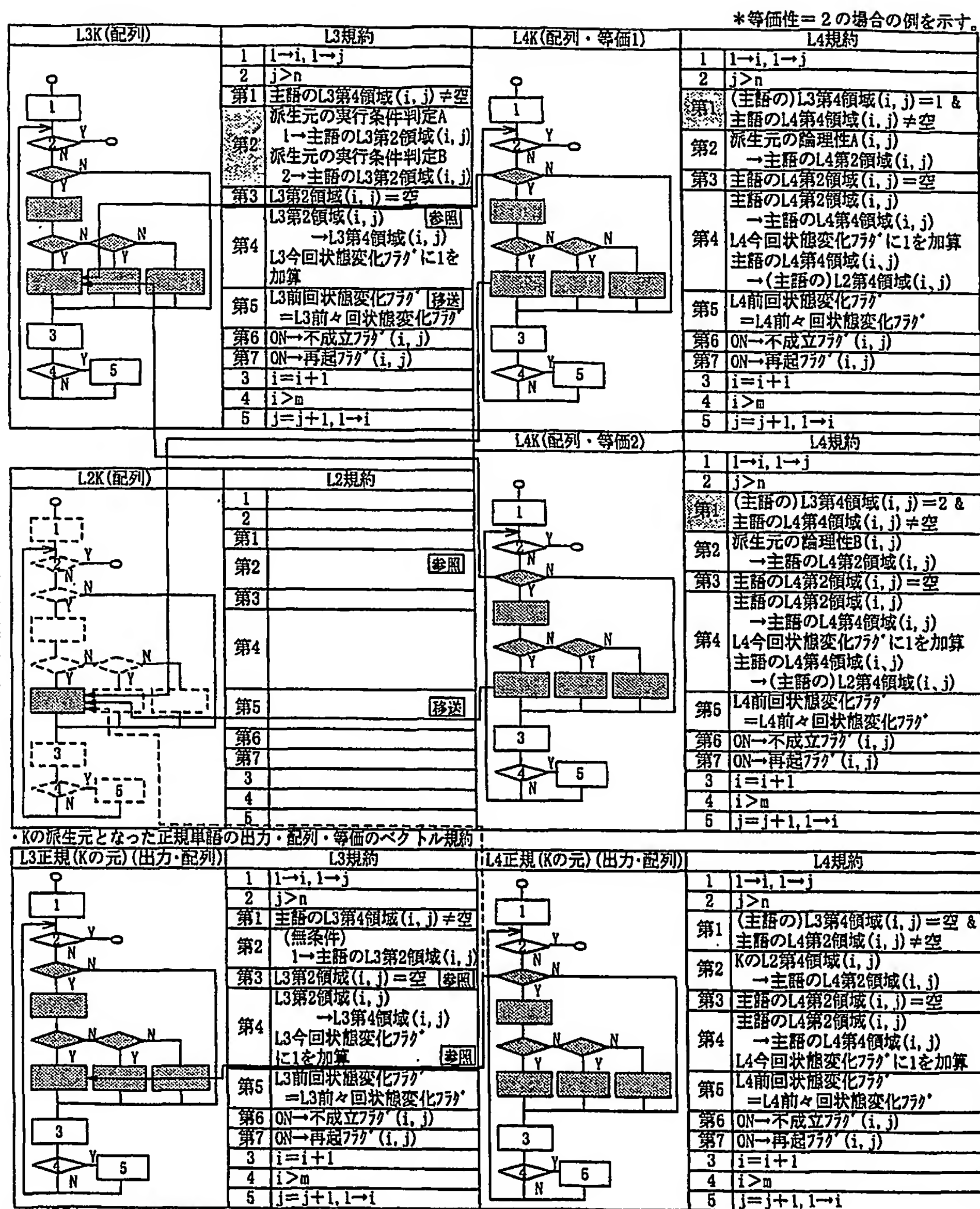
定義規則 1 2 : 主語がKでその属性が等価のベクトル

[図43]



定義規則 1 3 : 主語がKでその属性が配列のベクトル

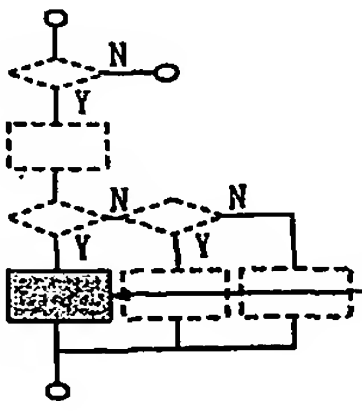
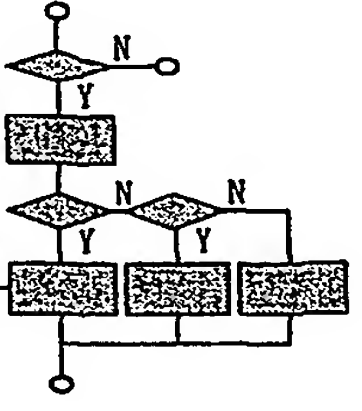
[図44]



定義規則14: 主語がKでその属性が等価、配列のベクトル



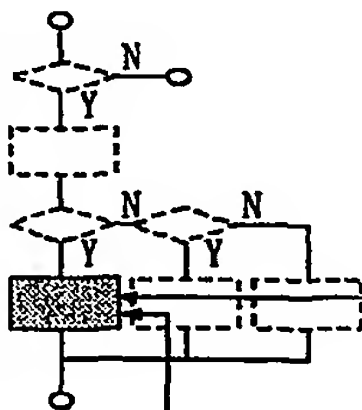
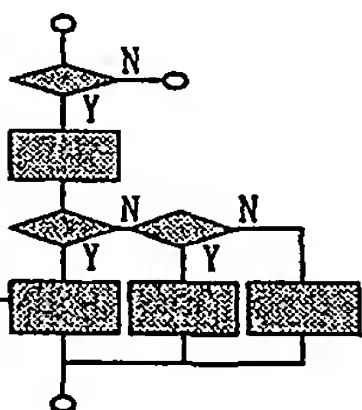
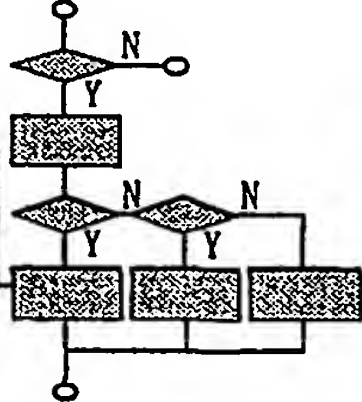
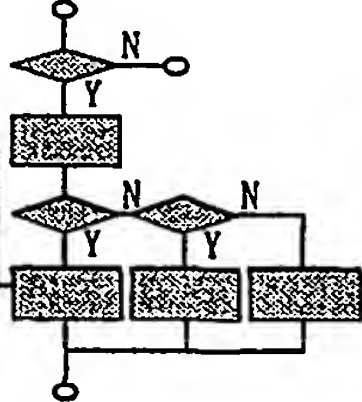
[図45]

L2M	L2規約		L4M(出力)	L4規約	
	第1			第1	主語のL4第4領域≠空
	第2			第2	実行条件判定&主語の論理性 →主語のL4第2領域
	第3			第3	主語のL4第2領域=空
	第4			第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域
	第5			第5	L4前回状態変化フラグ =L4前々回状態変化フラグ
	第6			第6	ON→不成立フラグ
	第7			第7	ON→再起フラグ

定義規則 15 : 主語がMでその属性が出力のベクトル

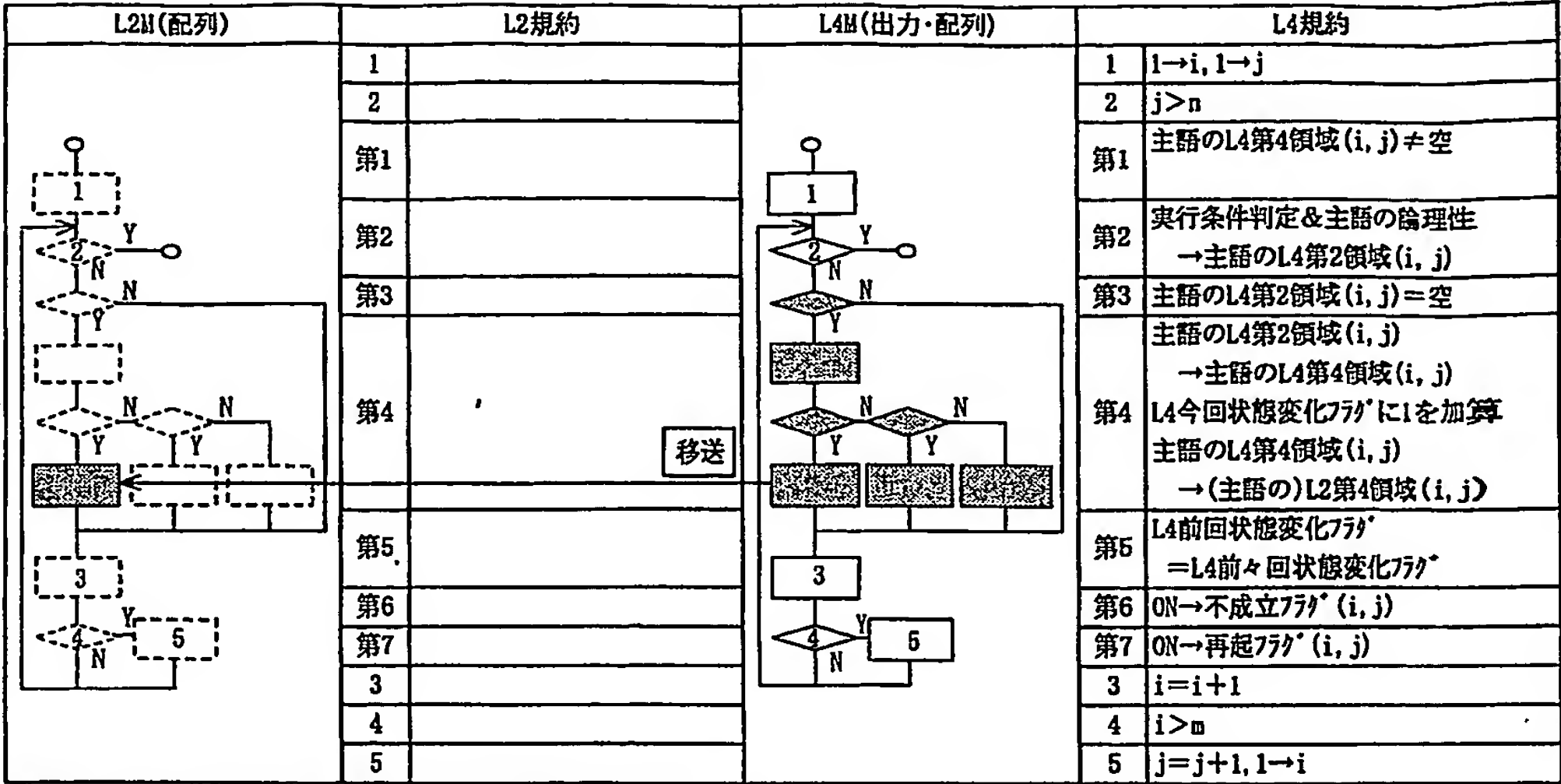
[図46]

\* 等価性 = 2 の場合の例を示す。

L2M	L2規約		L4M(出力・等価1)	L4規約		
	第1			第1	主語のL4第4領域≠空	
	第2			第2	実行条件判定A&主語の論理性A →主語のL4第2領域	
	第3			第3	主語のL4第2領域=空	
	第4			第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域	
	第5			第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
	第6			第6	ON→不成立フラグ	
	第7			第7	ON→再起フラグ	
			L4M(出力・等価2)	L4規約		
				第1	主語のL4第4領域≠空	
				第2	実行条件判定B&主語の論理性B →主語のL4第2領域	
				第3	主語のL4第2領域=空	
				第4	主語のL4第2領域 →主語のL4第4領域 L4今回状態変化フラグに1を加算 主語のL4第4領域 →(主語の)L2第4領域	
				第5	L4前回状態変化フラグ =L4前々回状態変化フラグ	
				第6	ON→不成立フラグ	
				第7	ON→再起フラグ	

定義規則 16 : 主語がMでその属性が出力、等価のベクトル

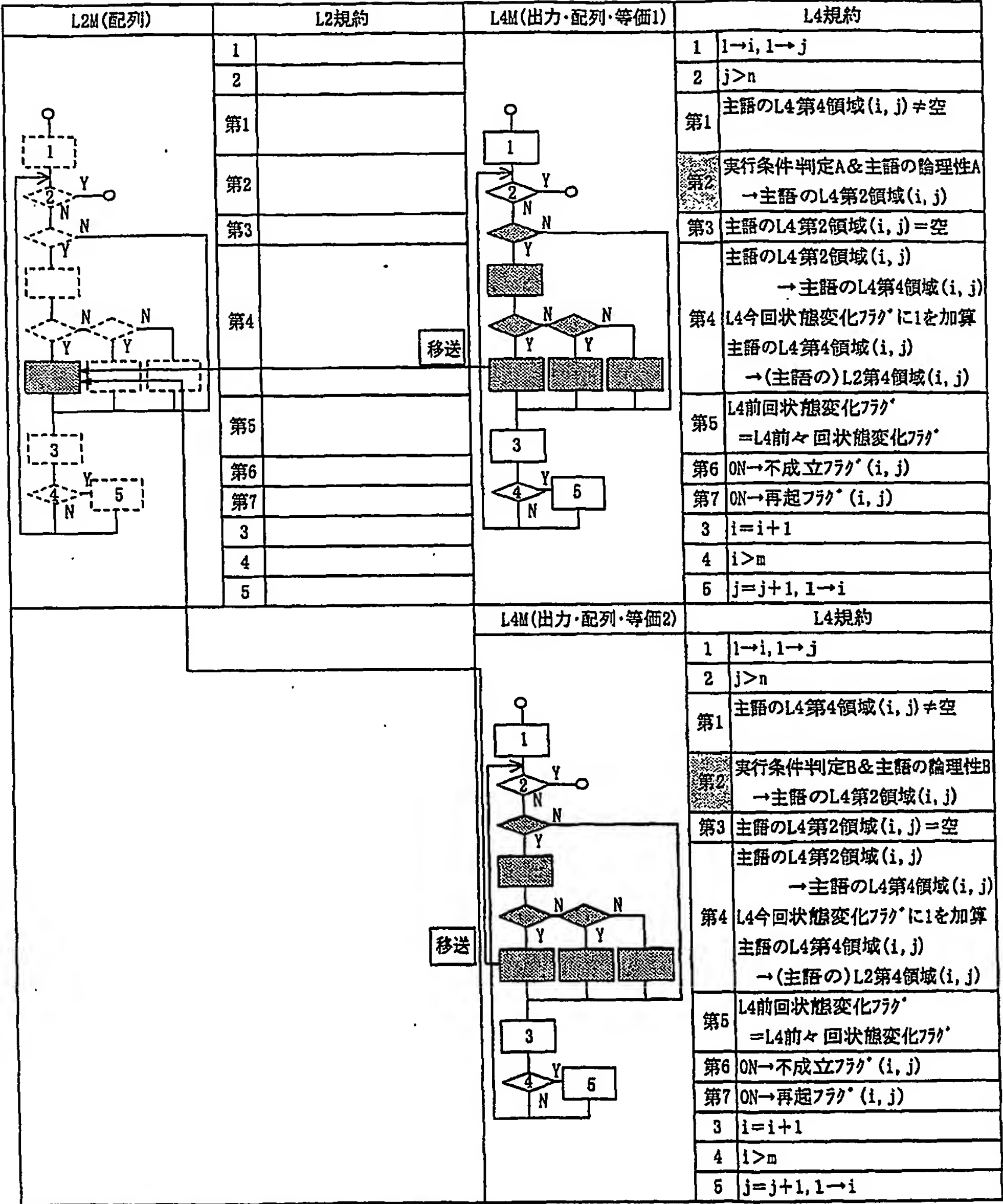
[図47]



定義規則 17 : 主語がMでその属性が出力, 配列のベクトル

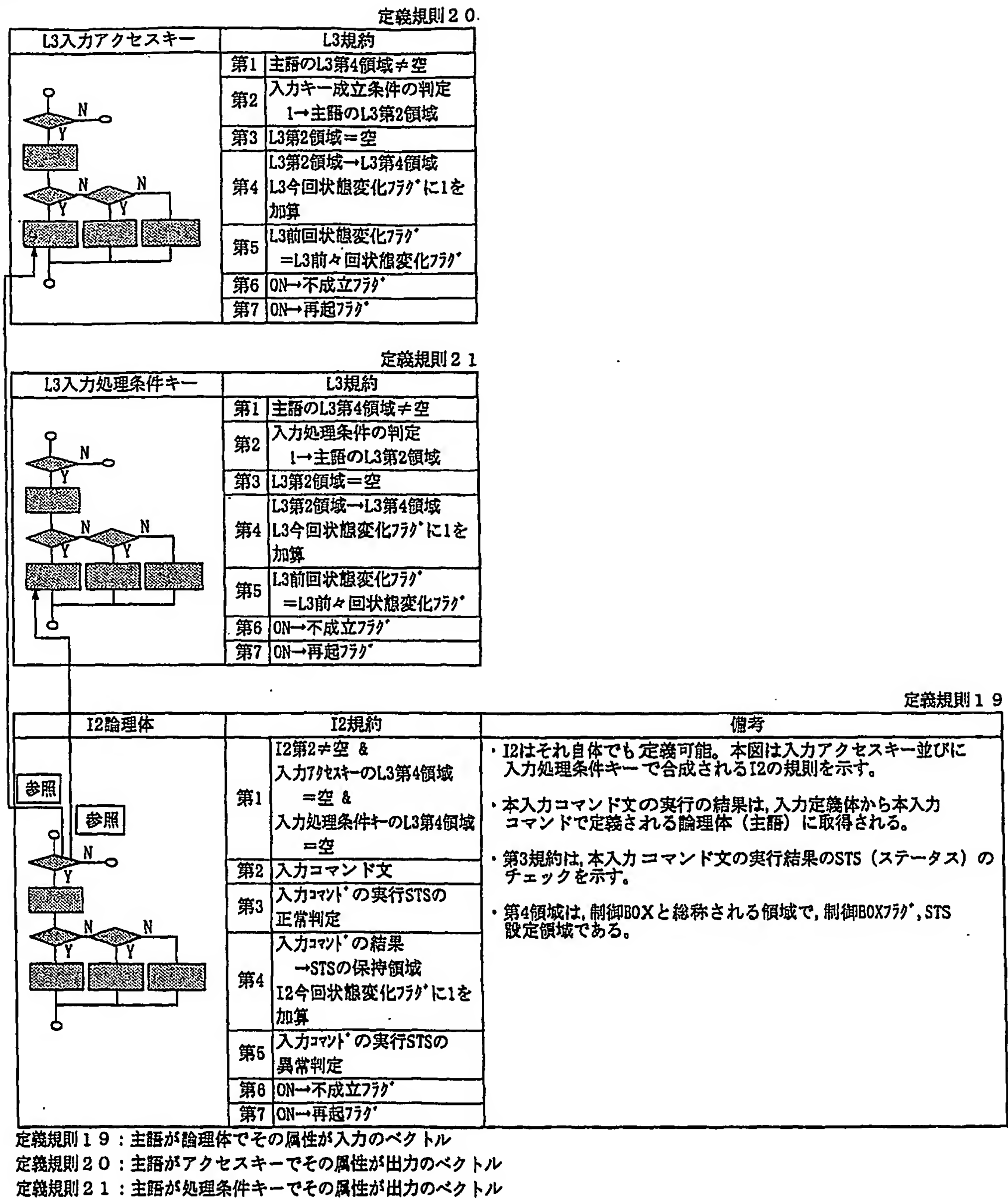
[図48]

\*等価性=2の場合の例を示す。

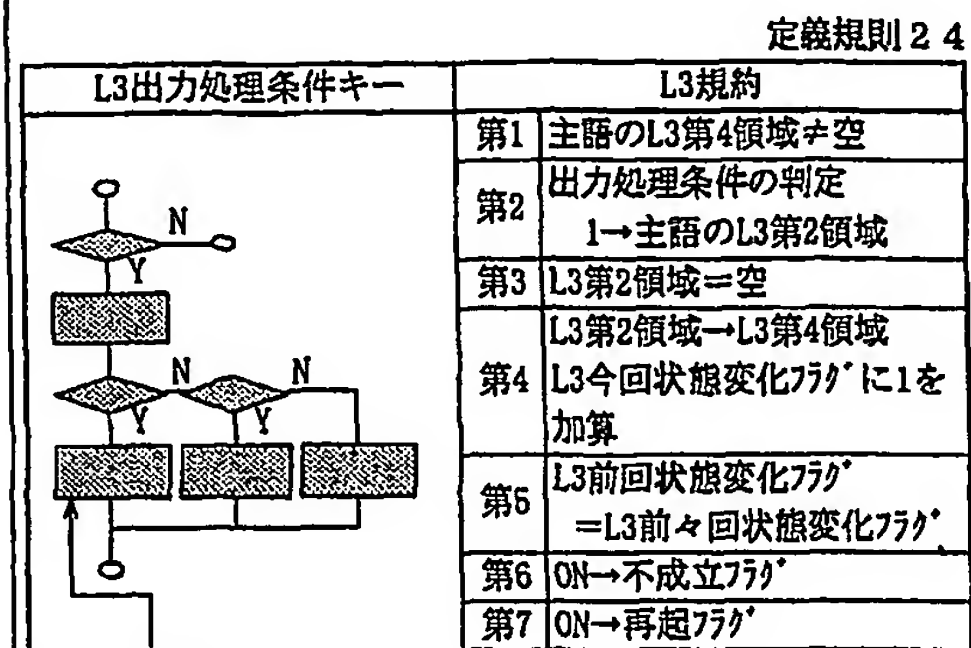
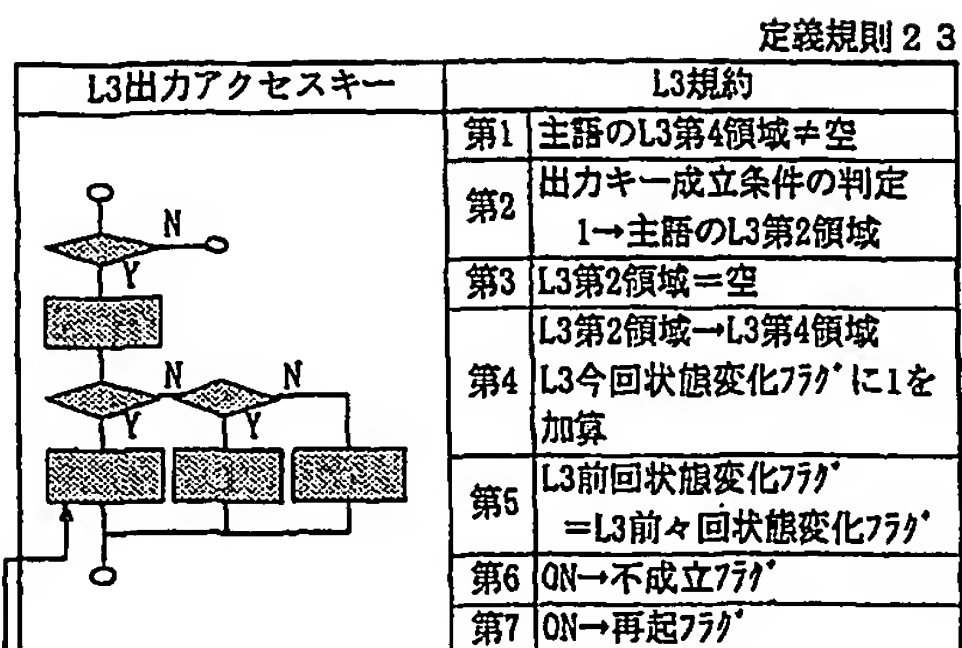


定義規則18：主語がMでその属性が出力，等価，配列のベクトル

[図49]



[図50]



04論理体		04規約	備考
<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;">参照</div> </div>	第1	04第4の出力済フラグ ≠ 空 & 出力アクセスのL3第4領域 = 空 & 出力処理条件キーのL3第4領域 = 空	<ul style="list-style-type: none"> <li>・ 04はそれ自体でも定義可能。本図は出力アクセスキー並びに出力処理条件キーで合成される04の規則を示す。</li> <li>・ 本出力コマンド文で使用する論理体は、主語のL4第4領域の集合である。</li> </ul>
	第2	出力コマンド文	<ul style="list-style-type: none"> <li>・ 第9規約は、本出力コマンド文の実行結果のSTS（ステータス）のチェックを示す。</li> </ul>
	第3	出力コマンドの実行STSの正常判定	
	第4	出力コマンドの結果 →STSの保持領域 04今回状態変化フラグに1を加算 経路制御TBLの04成立情報に発報	<ul style="list-style-type: none"> <li>・ 第4領域は、制御BOXと総称される領域で、制御BOXフラグ、STS設定領域である。</li> </ul>
	第5	出力コマンドの実行STSの異常判定	
	第6	ON→不成立フラグ	
	第7	ON→再起フラグ	

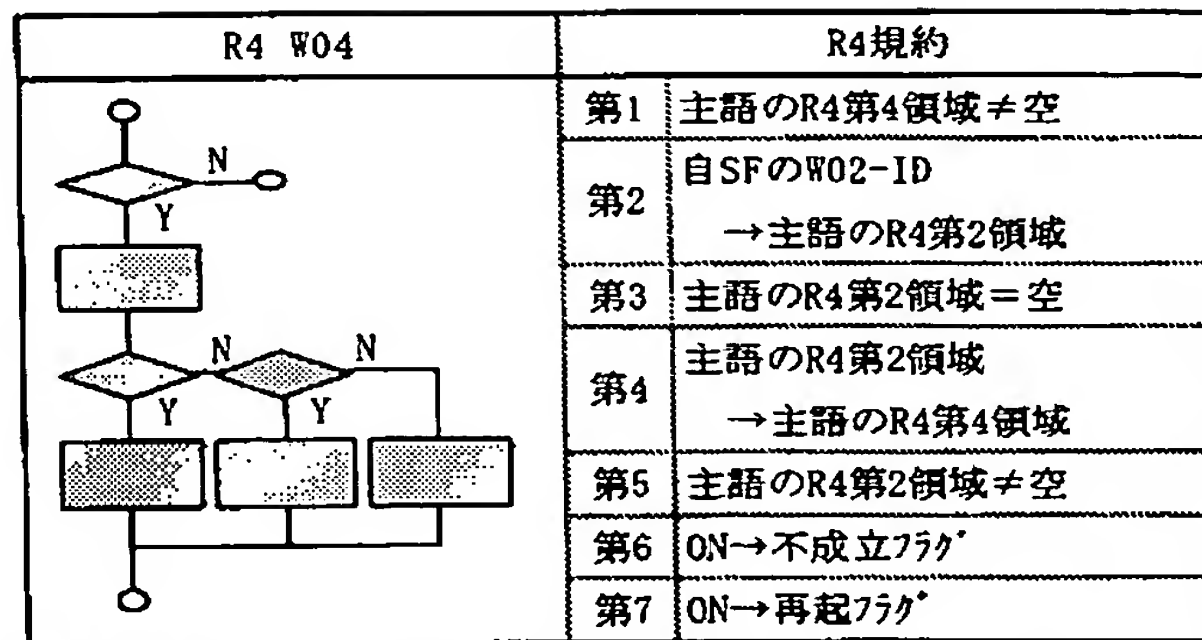
**定義規則 2.2 :** 主語が論理体でその属性が出力のベクトル

定義規則 23 : 主語がアクセスキーでその属性が出力のベクトル

定義規則 24: 主語が処理条件キーでその属性が出力のベクトル

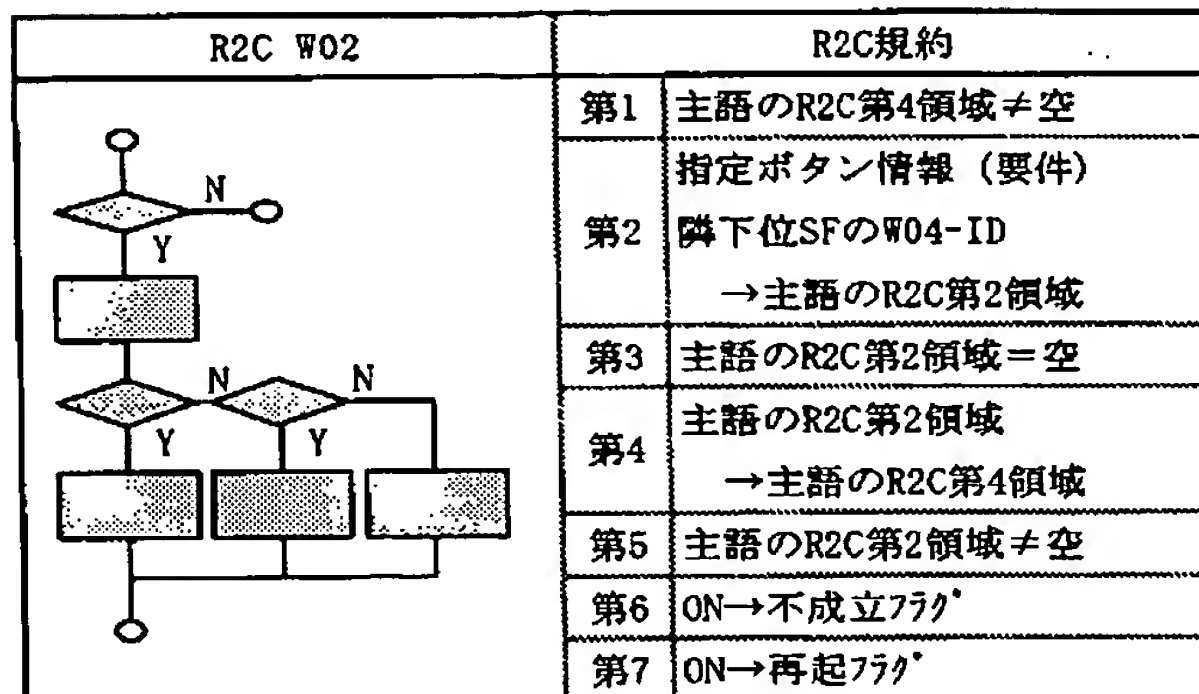


[図51]



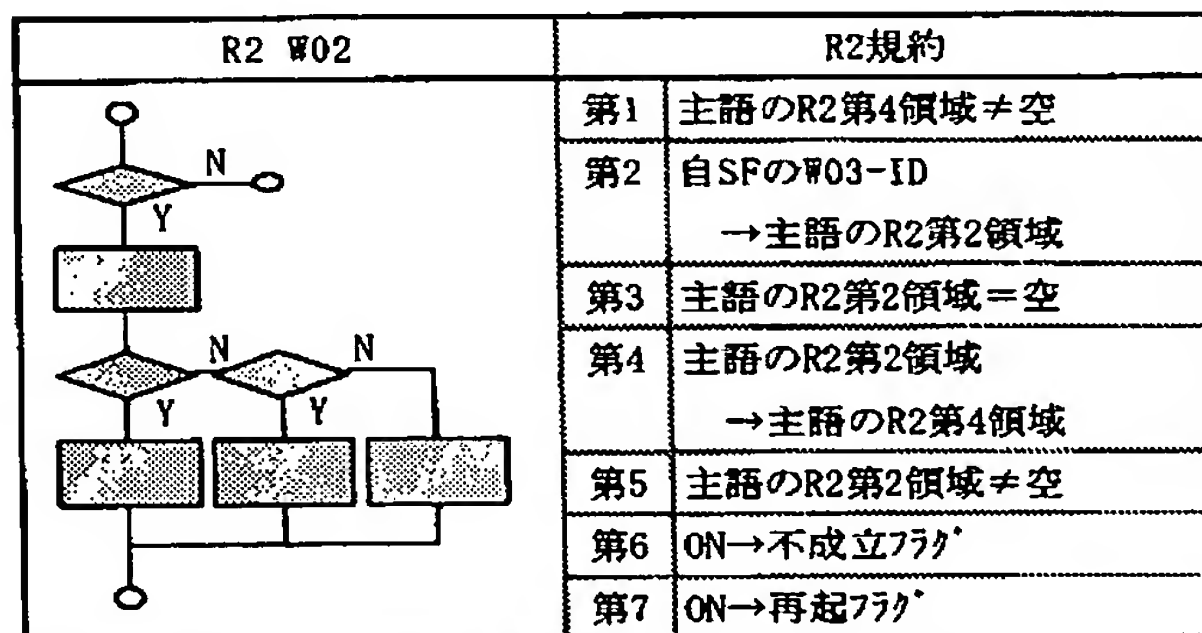
定義規則 2 5 : 主語がパレットW04でR4のベクトル

[図52]



定義規則 2 6 : 主語がパレットW02でR2Cのベクトル

[図53]



定義規則 2 7 : 主語がパレットW02でR2のベクトル

[図54]

R3R W03	R3R規約
	第1 主語のR3R第4領域≠空
	第2 経路判定条件 自SFのW04-ID →主語のR3R第2領域
	第3 主語のR3R第2領域=空
	第4 主語のR3R第2領域 →主語のR3R第4領域
	第5 主語のR3R第2領域≠空
	第6 ON→不成立フラグ
	第7 ON→再起フラグ

定義規則 2 8 : 主語がパレットW03でR3Rのベクトル

[図55]

R3E W03	R3E規約
	第1 主語のR3E第4領域≠空
	第2 SF終了条件(業務条件) ON→主語のR3E第2領域
	第3 主語のR3E第2領域=空
	第4 主語のR3E第2領域 →主語のR3E第4領域 主語のR3E第4領域 →経路制御TBLのR3E情報
	第5 主語のR3E第2領域≠空
	第6 ON→不成立フラグ
	第7 ON→再起フラグ

定義規則 2 9 : 主語がパレットW03でR3Eのベクトル

[図56]

R3C W03	R3C規約
	第1 主語のR3C第4領域≠空
	第2 経路判定条件 隣下位SFのW04-ID →主語のR3C第2領域
	第3 主語のR3C第2領域=空
	第4 主語のR3C第2領域 →主語のR3C第4領域
	第5 主語のR3C第2領域≠空
	第6 ON→不成立フラグ
	第7 ON→再起フラグ

定義規則 3 0 : 主語がパレットW03でR3Cのベクトル

[図57]

R3D W03	R3D規約	
	第1	主語のR3D第4領域≠空
	第2	経路判定条件 隣上位SFのW03-ID →主語のR3D第2領域
	第3	主語のR3D第2領域=空
	第4	主語のR3D第2領域 →主語のR3D第4領域
	第5	主語のR3D第2領域≠空
	第6	ON→不成立フラグ
	第7	ON→再起フラグ

定義規則 3 1：主語がパレットW03でR3Dのベクトル

[図58]

R3M W03	R3M規約	
	第1	主語のR3M第4領域≠空
	第2	経路判定条件 隣々上位SFのW04-ID →主語のR3M第2領域
	第3	主語のR3M第2領域=空
	第4	主語のR3M第2領域 →主語のR3M第4領域
	第5	主語のR3M第2領域≠空
	第6	ON→不成立フラグ
	第7	ON→再起フラグ

定義規則 3 2：主語がパレットW03でR3Mのベクトル

[図59]

S4, I2第2領域	S4規約	
	第1	主語のL2第4領域≠空
	第2	主語のI2第2領域をクリア
	第3	NOP
	第4	NOP
	第5	NOP
	第6	NOP
	第7	NOP

定義規則 3 3：主語がI2第2領域のS4のベクトル

[図60]

S4, I2第4領域	S4規約	
	第1	NOP
	第2	主語のI2第4領域をクリア
	第3	NOP
	第4	NOP
	第5	NOP
	第6	NOP
	第7	NOP

定義規則 3 4 : 主語が I 2 第 4 領域の S 4 のベクトル

[図61]

S4, IAK, 主語のL3第4領域	S4規約	
	第1	04第4領域の出力済フラグ = 空
	第2	主語のL3第4領域をクリア
	第3	NOP
	第4	NOP
	第5	NOP
	第6	NOP
	第7	NOP

定義規則 3 5 : 主語が入力アクセスキーの L 3 第 4 領域の S 4 のベクトル

[図62]

S4, ICK, 主語のL3第4領域	S4規約	
	第1	04第4領域の出力済フラグ = 空
	第2	主語のL3第4領域をクリア
	第3	NOP
	第4	NOP
	第5	NOP
	第6	NOP
	第7	NOP

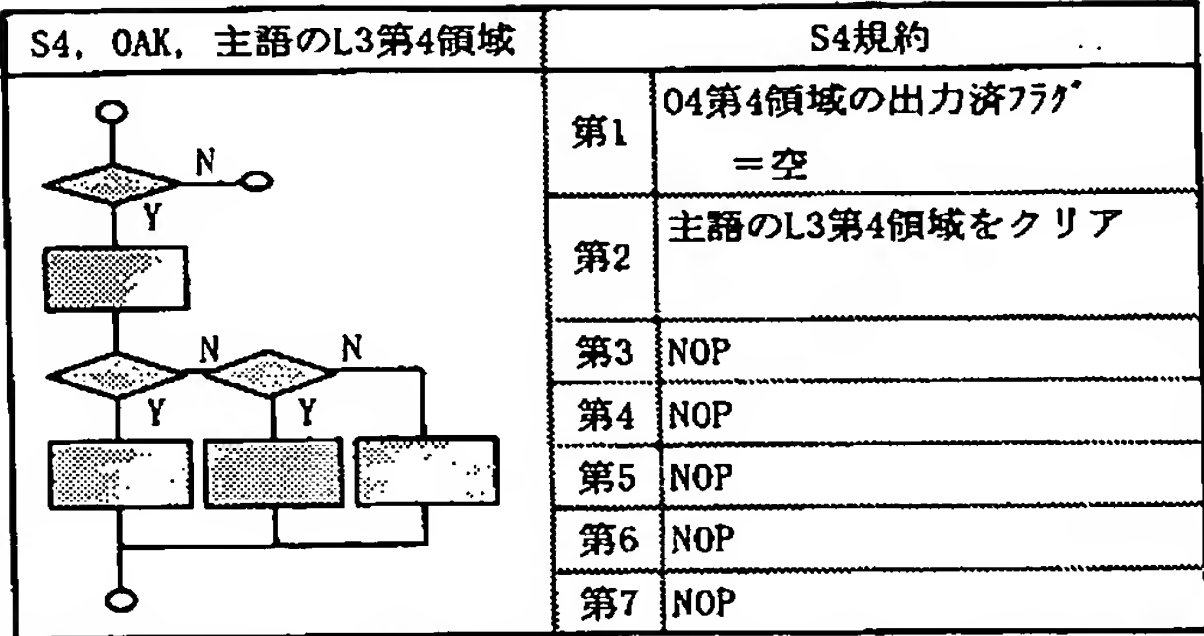
定義規則 3 6 : 主語が入力処理条件キーの L 3 第 4 領域の S 4 のベクトル

[図63]

S4, O4第4領域	S4規約	
	第1	NOP
	第2	主語のO4第4領域をクリア
	第3	NOP
	第4	NOP
	第5	NOP
	第6	NOP
	第7	NOP

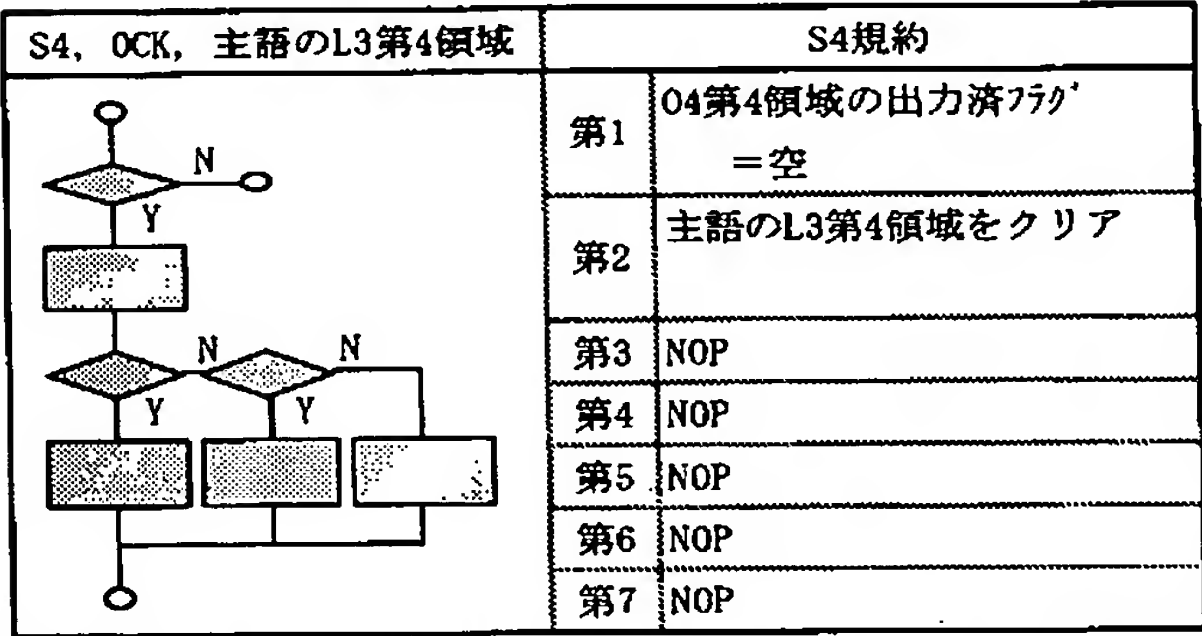
定義規則 3 7 : 主語が O 4 第 4 領域の S 4 のベクトル

[図64]



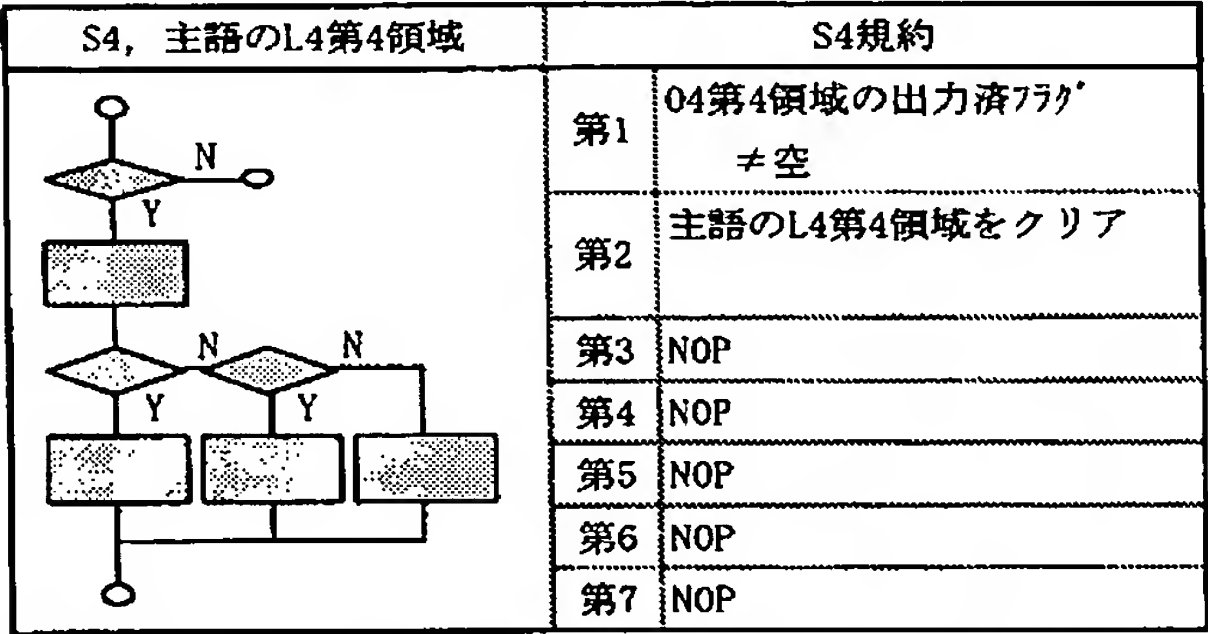
定義規則 3 8 : 主語が出力アクセスキーの L 3 第 4 領域の S 4 のベクトル

[図65]



定義規則 3 9 : 主語が出力処理条件キーの L 3 第 4 領域の S 4 のベクトル

[図66]

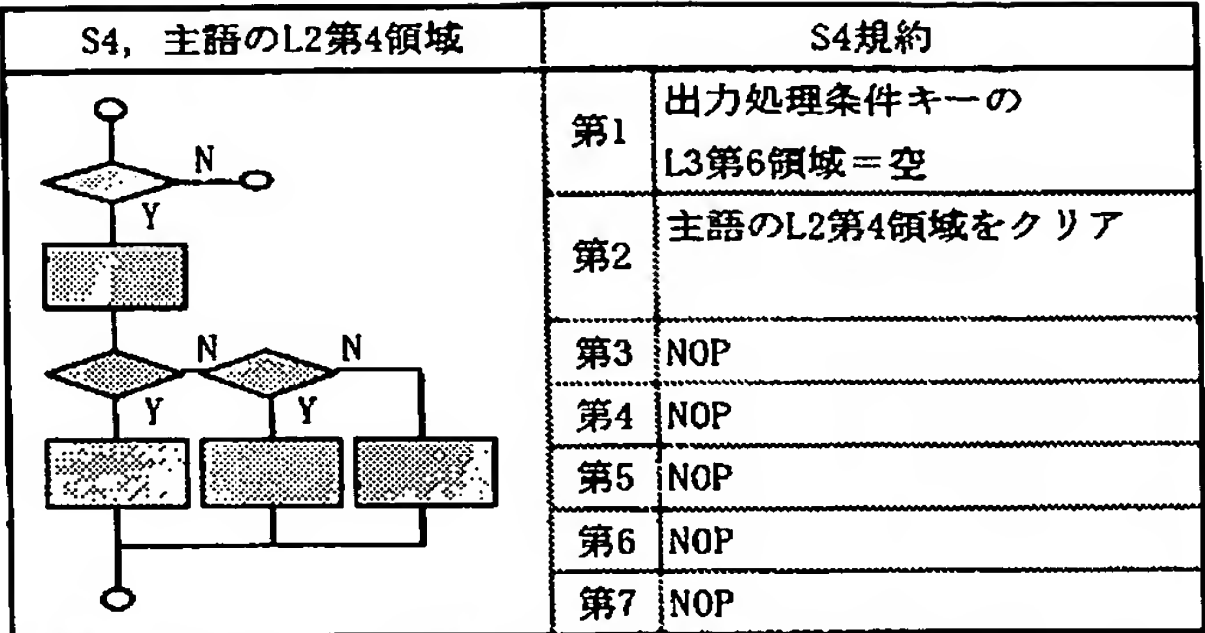


定義規則 4 0 : 主語が L 4 第 4 領域の S 4 のベクトル

定義規則 4 1 から 5 1 については定義規則 4 0 参照

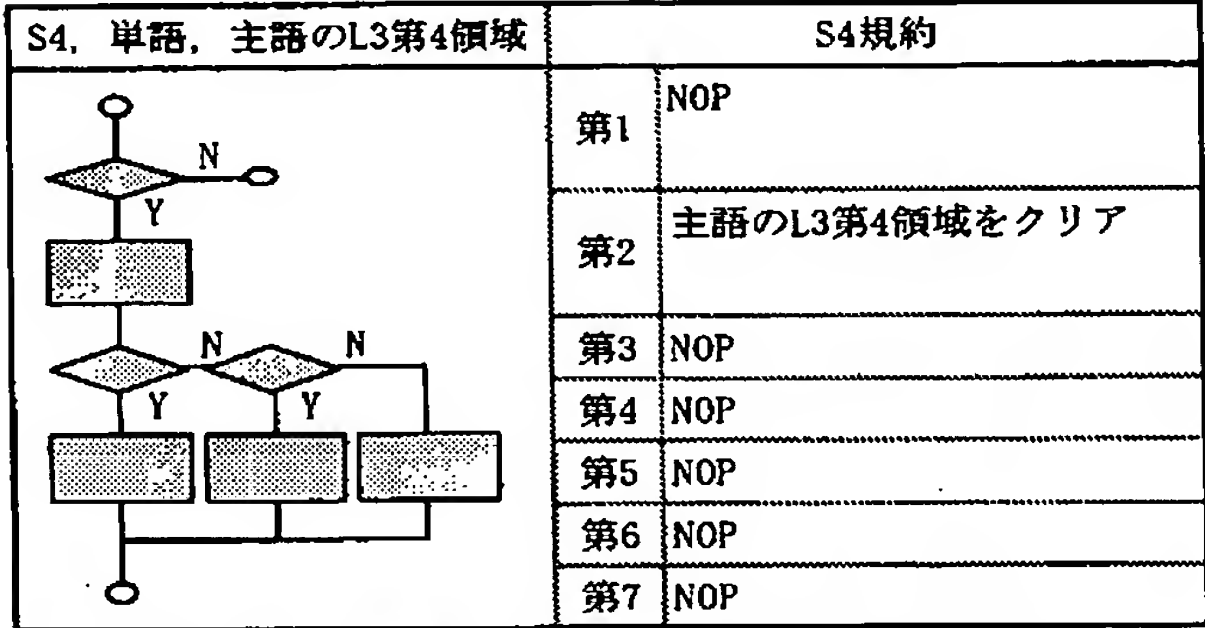


[図67]



定義規則 5 2 : 主語が L 2 第 4 領域の S 4 のベクトル  
定義規則 5 3 から 5 7 については定義規則 5 2 参照

[図68]



定義規則 5 8 : 主語が L 3 第 4 領域の S 4 のベクトル  
定義規則 5 9 から 7 3 については定義規則 5 8 参照

## [図69]

定義規則1で定義されるパクトルL2（正規） - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB1N2
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB1N2
Private 0%310_0%280_0%290_0%020 As 0%230
Rem PUB5C2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6P2
Private CTRL_W0%300_0%310_0%280_0%290_0%020 NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T2
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%220
Rem PRV7R2
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem $PRV2W2
0%1620
Rem $PRV2E2
Rem $PRV4W2
0%1550
Rem $PRV4E2
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L2_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020 - CNS NOT KUH 0%230 Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%150
BOX_3:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020_wk <> CNS NOT KUH 0%230 Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG

```

## [図70]

定義規則1で定義されるベクトル1.2 (正規) - 2

```

W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_INC_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```

[図71]

定義規則2で定義されるベクトルL2（正規） - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB1N2
Private W%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB1N2
Private 0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB5C2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F2
Private CTRL_W%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem #DEF PRV
Rem PRV2T2
Private W%300_0%310_0%280_0%290_0%020_vk(0%260,0%270) As 0%230
Rem PRV7R2
Private CTRL_W%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Private Gyou as Integer
Private Rctu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W2
0%1520
Rem $PRV2E2
Rem $PRV4V2
0%1550
Rem $PRV4E2
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L2_0%310_0%280_0%290_0%020()
    Gyou = 0
    Rctu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then

```

[図72]

定義規則 2 で定義されるベクトル L 2 (正規) - 2

```

Rem PRVLG
    Uti_Max = 0%270
Rem PRVLG
    Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS.NOT_KUH_0%230 Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0%150
BOX_3:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS.NOT_KUH_0%230 Then
            GoTo BOX_4
        End If
        GoTo BOX_5
BOX_4:
Rem PRVLG
        W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
        CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
        GoTo BOX_F
BOX_5:
Rem PRVLG
        If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
            - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7

```



## [図73]

定義規則2で定義されるベクトルL 2 (正規) - 3

```

BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PCEND

```

[図74]

定義規則3で定義されるベクトルL3（正規） - 1

```

Option Explicit
Rem #DEFPOB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_0%020 As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%260_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB6R3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB0F3
Private CTRL_W0%300_0%310_0%280_0%290_0%020 NOT VALID FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_0%020_wk As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020 < CNS_NOT_KUH_Integer Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_Integer Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG

```

## [図75]

定義規則 3 で定義されるベクトル L 3 (正規) - 2

```

      W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
        = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG PP Then
        GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```

[図76]

定義規則 3 で定義されるベクトル L 4 (正規) - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB4N4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB6Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEF END
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEF PRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUB KU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEF END
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem PRVLG

```

[図77]

定義規則3で定義されるベクトルL4 (正規) - 2

```

      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%230 And _
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%230 Then Go To Box_3
Rem PRVLG
      0%170
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%290_0%290_0%020_wk
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX E:
End Sub
Rem *** PRCEND

```



[図78]

定義規則4で定義されるベクトル1.3 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4NS
Private W%300_0%310_0%280_0%290_0%020 As Integer
Rem PUBINS
Private W%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W%300_0%310_0%280_0%290_0%020_NUT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W%300_0%310_0%280_0%290_0%020_wk0%070 As Integer
Rem PRV7R3
Private CTRL_W%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%1600
Rem $PRV2E3
Rem $PRV4W3
0%1600
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If W%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_Integer Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PUBL3F
If W%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then GoTo BOX_4
GoTo BOX_5
BOX_4:
Rem PUBL3F
If W%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then W%300_0%310_0%280_0%290_0%020 = 0%070
Rem PRVLG

```

[図79]

定義規則4で定義されるベクトル1.3 (正規) - 2

```
CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG = CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG + 1
    GoTo BOX_E
BOX_5:
Rem PRVLG
    If CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG_P
    = CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG_PP Then
        GoTo BOX_6
    End If
    GoTo BOX_7
BOX_6:
Rem PRVLG
    CTRL_W0X300_0X310_0X280_0X290_0X020_NOT_VALID_FLG = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W0X300_0X290_ING_FLG = True
    GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND
```

## [図80]

定義規則 4 で定義されるベクトル L 4 (正規) - 1

```

Option Explicit
Rem #DEFPU8
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PURIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem PUR4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PURKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2E4
0%1720
Rem $PRV2E4
Rem $PRV4E4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L1_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem PRVLC

```

[図81]

定義規則4で定義されるバクトルL 4 (正規) - 2

```

      If W%X300_0%X310_0%X280_0%X290_0%X020 = CNS_NOT_KUH_0%X230 And _
Rem PRVLG
      W3_0%X310_0%X280_0%X290_0%X020 = 0%X070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W%X3050_0%X3150_0%X2850_0%X2950_0%X2000 = CNS_NOT_KUH_0%X2350 Then Go To Box_3
Rem PRVLG
      0%X170
BOX_3:
Rem PRVLG
      If W%X300_0%X310_0%X280_0%X290_0%X020_wk <> CNS_NOT_KUH_0%X230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W%X300_0%X310_0%X280_0%X290_0%X020 = W%X300_0%X310_0%X280_0%X290_0%X020_wk
Rem PRVLG
      CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P
      = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W%X300_0%X290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```

## [図82]

定義規則5で定義されるベクトルL3 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
Private W0%300_0%310_0%260_0%290_0%020(0%260,0%270) As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260,0%270) As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then

```



## [図83]

定義規則5で定義されるベクトルL3（正規） - 2

```

Rem PRVLG
    Uti_Max = 0%270
Rem PRVLG
    Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0%160
BOX_3:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS_NOT_KUH_Integer Then
            GoTo BOX_4
        End If
        GoTo BOX_5
BOX_4:
Rem PRVLG
        W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
        CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
            - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7

```

[図84]

定義規則5で定義されるベクトルL3（正規） - 3

```

BOX_6:
Rem PRVLG
    CTRL_W%X300_@X310_@X280_@X290_@X020_NOT_VALID_FLG(Gyou, Retu) = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W%X300_@X290_ING_FLG = True
    GoTo BOX_E
BOX_E:
    Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
    If @X340 < @X350 then
        Gyou = Uti_Cnt
        Retu = Soto_Cnt
    End If
Rem PRVLG
    If @X340 > @X350 then
        Gyou = Soto_Cnt
        Retu = Uti_Cnt
    End If
    Loop
    Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
    If @X340 < @X350 then
        Gyou = Uti_Cnt
        Retu = Soto_Cnt
    End If
Rem PRVLG
    If @X340 > @X350 then
        Gyou = Soto_Cnt
        Retu = Uti_Cnt
    End If
    Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PRCEND

```

[図85]

定義規則 5 で定義されるベクトル L 4 (正規) - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB4N1
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB3N4
Private W0%3050_0%3150_0%2950_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%230_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260,0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260,0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEF PRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260,0%270) As 0%230
Rem PRV7K4
Private CTRL_W0%300_0%290_INC_FLG As Boolean
Rem #PUB KU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Private Gyou As Integer
Private Retu As Integer
Private Uti_Cnt As Integer
Private Soto_Cnt As Integer
Private Uti_Max As Integer
Private Soto_Max As Integer
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```

## [図86]

定義規則5で定義されるベクトルL4（正規） - 2

```

Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
        End If
Rem PRVLG
        If 0%340 > 0%350 then
Rem PRVLG
            Uti_Max = 0%270
Rem PRVLG
            Soto_Max = 0%260
            End If
            Uti_Cnt = 0
            Soto_Cnt = 0
            Do Until Soto_Cnt = Soto_Max + 1
                Uti_Cnt = 0
Rem PRVLG
                If 0%340 < 0%350 then
                    Gyou = Uti_Cnt
                    Retu = Soto_Cnt
                End If
Rem PRVLG
                If 0%340 > 0%350 then
                    Gyou = Soto_Cnt
                    Retu = Uti_Cnt
                End If
                Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
                If 0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_0%230 And _
Rem PRVLG
                0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
                    GoTo BOX_2
                End If
                GoTo BOX_F
BOX_2:
Rem PUBIN
                If 0%3050_0%3150_0%2850_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
                0%170

```

## [図87]

定義規則5で定義されるベクトルL 4 (正規) - 3

```

BOX_3:
Rem PRVLG
  If W0%300_0%310_0%280_0%290_0%020 wk(Gyou, Retu) <> CNS_NOT_KU1_0%220 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020 wk(Gyou, Retu)
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then

```



[図88]

定義規則5で定義されるベクトルL4 (正規) - 4

```
Gyou = Soto_Cnt  
Retu = Uti_Cnt  
  
End If  
  
Loop  
  
Gyou = 0  
Retu = 0  
Uti_Cnt = 0  
Soto_Cnt = 0  
Uti_Max = 0  
Soto_Max = 0  
End Sub  
Rem *** PROCEND
```

## [図89]

定義規則6で定義されるベクトルL3（正規）－1

```

Option Explicit
Rem #DEFPIR
Rem PUB4K3
Private W%300_0%310_0%250_0%290_0%020(0%260,0%270) As Integer
Rem PUBIN3
Private W%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEPEND
Rem #DEFPRV
Rem PRV2T3
Private W%300_0%310_0%280_0%290_0%020_wk0%070(0%260,0%270) As Integer
Rem PRV7R3
Private CTRL_W%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEPEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then

```

[図90]

定義規則6で定義されるベクトルL3 (正規) - 2

```

Rem PRVLG
    Uti_Max = 0%270
Rem PRVLG
    Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max - 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0%160
BOX_3:
Rem PUBL3F
        If W0%300_0%310_0%280_0%290_0%020_wk0%070(Gyou, Retu) = 1 Then GoTo BOX_4
        GoTo BOX_5
BOX_4:
Rem PUBL3F
        If W0%300_0%310_0%280_0%290_0%020_wk0%070(Gyou, Retu) = 1
            Then W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070
Rem PRVLG
        CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
            CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7
BOX_6:

```

## [図91]

定義規則6で定義されるベクトルL3（正規）－3

```

Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou,Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt - 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PCEND

```

[図92]

定義規則6で定義されるベクトルL4（正規） - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB1N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB1N4
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020 NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260,0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260,0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEF PRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260,0%270) As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Private Gyou As Integer
Private Retu As Integer
Private Uti_Cnt As Integer
Private Soto_Cnt As Integer
Private Uti_Max As Integer
Private Soto_Max As Integer
Rem $PRV2E4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```

## [図93]

定義規則6で定義されるベクトルL4 (正規) - 2

```

Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then
Rem PRVLG
        Uti_Max = 0%270
Rem PRVLG
        Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max - 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_0%230 And _
Rem PRVLG
        W3_0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PUBIN
        If W0%3050_0%3100_0%2850_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
        0%170

```



## [図94]

定義規則6で定義されるベクトルL 4 (正規) - 3

```

BOX_3:
Rem PRVLG
  If W%300_0%310_0%280_0%290_0%020 wk(Gyou, Retu) <> CNS_NOT_KUH_0%230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5

BOX_4:
Rem PRVLG
  W%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
  CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E

BOX_5:
Rem PRVLG
  If CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_FF Then
    GoTo BOX_6
  End If
  GoTo BOX_7

BOX_6:
Rem PRVLG
  CTRL_W%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E

BOX_7:
Rem PRVLG
  CTRL_W%300_0%290_ING_FLG = True
  GoTo BOX_E

BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then

```

[図95]

定義規則6で定義されるベクトルL 4 (正規) - 4

```
Gyou = Soto_Cnt  
Retu = Uti_Cnt  
End If  
Loop  
Gyon = 0  
Retu = 0  
Uti_Cnt = 0  
Soto_Cnt = 0  
Uti_Max = 0  
Soto_Max = 0  
End Sub  
Rem *** PRCEND
```

[図96]

定義規則7で定義されるベクトルL3 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_0%020 As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_0%020_wk As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2WS
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VPCREP Private Sub L3_0%310_0%280_0%250_0%020()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_Integer Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_Integer Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG

```

[図97]

定義規則 7 で定義されるベクトル L 3 (正規) - 2

```

W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%290_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_INC_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```

[図98]

定義規則 7 で定義されるベクトル L 4 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB1N4
Private W0%3050_0%3150_0%2350_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH String As String
Private CNS_NOT_KUH Integer As Integer
Private CNS_NOT_KUH Boolean As Boolean
Private CNS_NOT_KUH Long As Long
Private CNS_NOT_KUH Single As Single
Private CNS_NOT_KUH Double As Double
Private CNS_NOT_KUH Variant As Variant
Private CNS_NOT_KUH Currency As Currency
Private CNS_NOT_KUH Byte As Byte
Private CNS_NOT_KUH Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem PRVLG

```

[図99]

定義規則7で定義されるベクトルL 4 (正規) - 2

```

      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%230 And
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%230 Then Go To Box_3
Rem PRVLG
      0%170
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_E
BOX_4:
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX E:
End Sub
Rem *** PRCEND

```



[図100]

定義規則 8 で定義されるベクトル L 3 (正規) - 1

```

Option Explicit
Rem #DEFPIB
Rem PUB4N3
  Private W0%300_0%310_0%280_0%290_0%020 As Integer
Rem PUBIN3
  Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
  Private CTRL_W0%300_0%260_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
  Private CTRL_W0%300_0%310_0%280_0%290_0%020 NOT VALID FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
  Private W0%300_0%310_0%280_0%290_0%020_wk0%070 As Integer
Rem PRV7K3
  Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUB5KU
  Private CNS_NOT_KUH Integer As Integer
Rem SPRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
  BOX_1:
Rem PRVLG
    If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH Integer Then
      GoTo BOX_2
    End If
    GoTo BOX_E
  BOX_2:
Rem PRVLG
    0%160
  BOX_3:
Rem PUBL3F
    If W0%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then GoTo BOX_4
    GoTo BOX_5
  BOX_4:
Rem PUBL3F
    If W0%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then W0%300_0%310_0%280_0%290_0%020 = 0%070
Rem PRVLG

```

## [図101]

定義規則8で定義されるベクトルL3 (正規) - 2

```
CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
    GoTo BOX_E

BOX_5:
Rem PRVLG
    If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
        = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
        GoTo BOX_6
    End If
    GoTo BOX_7

BOX_6:
Rem PRVLG
    CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
    GoTo BOX_E

BOX_7:
Rem PRVLG
    CTRL_W0%300_0%290_ING_FLG = True
    GoTo BOX_E

BOX_E:
End Sub
Rem *** PRCEND
```

## [図102]

定義規則 8 で定義されるベクトル L 4 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C1
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y1
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_INC_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem PRVLC

```

## [図103]

定義規則8で定義されるベクトルL4 (正規) - 2

```

      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%230 And _
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%3050_0%3150_0%2850_0%2950_0%2000 = CNS_NOT_KUR_0%2350 Then Go To Box_3
Rem PRVLG
      0%170
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020 wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W0%300_0%310_0%250_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020 wk
Rem PRVLG
      W0%3000_0%3100_0%2800_0%2900_0%2000 = W0%300_0%310_0%280_0%290_0%020
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%230_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_INC_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

[図104]

定義規則9で定義されるベクトルL3 (正規) - 1

```

Option Explicit
Rem #DEFIUB
Rem PUB4N3
  Private W%X300_0%X310_0%X280_0%X290_0%X020(0%X260, 0%X270) As Integer
Rem PUBIN3
  Private W%X3050_0%X3150_0%X2850_0%X2950_0%X2000(0%X2650, 0%X2750) As 0%X2350
Rem PUB5C3
  Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
  Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
  Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
  Private CTRL_W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG(0%X260, 0%X270) As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
  Private W%X300_0%X310_0%X280_0%X290_0%X020_wk(0%X260, 0%X270) As Integer
Rem PRV7R3
  Private CTRL_W%X300_0%X290_1NG_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUUL Integer As Integer
  Private Gyou as Integer
  Private Retu as Integer
  Private Uti_Cnt as Integer
  Private Soto_Cnt as Integer
  Private Uti_Max as Integer
  Private Soto_Max as Integer
Rem $PRV2W3
0%X1620
Rem $PRV2E3
Rem $PRV4W3
0%X1650
Rem $PRV4E3
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub L3_0%X310_0%X280_0%X290_0%X020()
  Gyou = 0
  Retu = 0
Rem PRVLG
  If 0%X340 < 0%X350 then
Rem PRVLG
  Uti_Max = 0%X260
Rem PRVLG
  Soto_Max = 0%X270
  End If
Rem PRVLG
  If 0%X340 > 0%X350 then

```

## [図105]

定義規則 9 で定義されるベクトル L 3 (正規) - 2

```

Rem PRVLG
    Uti_Max = 0%270
Rem PRVLG
    Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0%160
BOX_3:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS_NOT_KUH_Integer Then
            GoTo BOX_4
        End If
        GoTo BOX_5
BOX_4:
Rem PRVLG
        W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
        CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG P
            = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7

```



## [図106]

定義規則 9 で定義されるベクトル L 3 (正規) - 3

```

BOX_6:
Rem PRVLG
  CTRL_F0%300_0%310_0%280_0%290_0%020 NOT VALID FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_F0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PRCEND

```

## [図107]

定義規則9で定義されるベクトルL 4 (正規) - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB4N4
Private W%X300_0%X310_0%X280_0%X290_0%X020(0%X260,0%X270) As 0%X230
Rem PUBIN4
Private W%X3050_0%X3150_0%X2850_0%X2950_0%X2000(0%X2650,0%X2750) As 0%X2350
Rem PUB5C4
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG(0%X260,0%X270) As Boolean
Rem #DEFEND
Rem PUB4R4
Private W%X3000_0%X3100_0%X2800_0%X2900_0%X2000(0%X260,0%X270) As 0%X230
Rem PUB4N3
Private W3_0%X310_0%X280_0%X290_0%X020(0%X260,0%X270) As Integer
Rem PUB4N2
Private W2_0%X310_0%X280_0%X290_0%X020(0%X260,0%X270) As 0%X230
Rem #DEFPRV
Rem PRV2T4
Private W%X300_0%X310_0%X280_0%X290_0%X020_wk(0%X260,0%X270) As 0%X230
Rem PRV7R4
Private CTRL W%X300_0%X290_ING_FLG As Boolean
Rem *PUBKUI
Private CNS_NOT_KUI String As String
Private CNS_NOT_KUI Integer As Integer
Private CNS_NOT_KUI Boolean As Boolean
Private CNS_NOT_KUI Long As Long
Private CNS_NOT_KUI Single As Single
Private CNS_NOT_KUI Double As Double
Private CNS_NOT_KUI Variant As Variant
Private CNS_NOT_KUI Currency As Currency
Private CNS_NOT_KUI Byte As Byte
Private CNS_NOT_KUI Date As Date
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W4
0%X1720
Rem $PRV2F4
Rem $PRV4W4
0%X1820

```

## [図108]

定義規則9で定義されるベクトルL 4 (正規) - 2

```

Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then
Rem PRVLG
        Uti_Max = 0%270
Rem PRVLG
        Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX 1:
Rem PRVLG
        If 0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_0%230 And
Rem PRVLG
            0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
                GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PUBIN
        If 0%3050_0%3150_0%2650_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
        0%170

```

## [図109]

定義規則 9 で定義されるベクトル L 4 (正規) - 3

```

BOX_3:
Rem PRVLG
  If W%X300_0%X310_0%X280_0%X290_0%X020_wk(Gyou, Retu) <> CNS_NOT_KUH_0%X230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W%X300_0%X310_0%X280_0%X290_0%X020(Gyou, Retu) = W%X300_0%X310_0%X280_0%X290_0%X020_wk(Gyou, Retu)
Rem PRVLG
  W%X3000_0%X3100_0%X2800_0%X2900_0%X0200(Gyou, Retu) = W%X300_0%X310_0%X280_0%X290_0%X020(Gyou, Retu)
Rem PRVLG
  CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P
    = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W%X300_0%X290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%X340 < 0%X350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%X340 > 0%X350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%X340 < 0%X350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If

```

## [図110]

定義規則9で定義されるベクトルL 4 (正規) - 4

```
Rem PRVLG
  If @%34@ > @%35@ then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PRCEND
```

[図111]

定義規則10で定義されるベクトルL3 (正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
  Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As Integer
Rem PUB1N3
  Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB6Y3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB6B3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
  Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
  Private W0%300_0%310_0%280_0%290_0%020_wk0%070(0%260,0%270) As Integer
Rem PRV7R3
  Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_Integer As Integer
  Private Gyou as Integer
  Private Retu as Integer
  Private Uti_Cnt as Integer
  Private Soto_Cnt as Integer
  Private Uti_Max as Integer
  Private Soto_Max as Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
  Gyou = 0
  Retu = 0
Rem PRVLG
  If 0%340 < 0%350 then
Rem PRVLG
  Uti_Max = 0%260
Rem PRVLG
  Soto_Max = 0%270
  End If
Rem PRVLG
  If 0%340 > 0%350 then

```



## [図112]

定義規則 10 で定義されるベクトル L3 (正規) ... 2

```

Rem PRVLG
    Uti_Max = 0x270
Rem PRVLG
    Soto_Max = 0x260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max - 1
        Uti_Cnt = 0
Rem PRVLG
        If 0x340 < 0x350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0x340 > 0x350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0x160
BOX_3:
Rem PUBL3F
        If W0x300_0x310_0x260_0x290_0x020_wk0x070(Gyou, Retu) = 1 Then GoTo BOX_4
        GoTo BOX_5
BOX_4:
Rem PUBL3F
        If W0x300_0x310_0x280_0x290_0x020_wk0x070(Gyou, Retu) = 1
            Then W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = 0x070
Rem PRVLG
        CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG - 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        If CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_P
            = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7
Rem PRVLG

```

## [図113]

定義規則 10 で定義されるベクトル L 3 (正規) - 3

```

IF CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
  = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
  GoTo BOX_6
End If
GoTo BOX_7
BOX_6:
Rem PRVLG
CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
GoTo BOX_E
BOX_7:
Rem PRVLG
CTRL_W0%300_0%290_ING_FLG = True
GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt - 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PRCEND

```

## [図114]

定義規則 10 で定義されるベクトル L 4 (正規) -- 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260, 0%270) As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650, 0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PT As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260, 0%270) As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260, 0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260, 0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260, 0%270) As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_WK(0%260, 0%270) As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PURKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```

## [図115]

定義規則10で定義されるベクトルL4（正規）－2

```

Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then
Rem PRVLG
        Uti_Max = 0%270
Rem PRVLG
        Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If 0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_0%230 And
Rem PRVLG
            0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
                GoTo BOX_2
            End If
            GoTo BOX_E
BOX_2:
Rem PUBIN
        If 0%3050_0%3150_0%2850_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
        0%170

```

## [図116]

定義規則 10 で定接されるベクトル L 4 (正規) - 3

```

BOX_3:
Rem PRVLG
  If W%X300_0%X310_0%X280_0%X290_0%X020_wk(Gyou, Retu) <> CHS_NOT_KUH_0%X230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W%X300_0%X310_0%X280_0%X290_0%X020(Gyou, Retu) = W%X300_0%X310_0%X280_0%X290_0%X020_wk(Gyou, Retu)
Rem PRVLG
  W%X3000_0%X3100_0%X2800_0%X2900_0%X0200(Gyou, Retu) = W%X300_0%X310_0%X280_0%X290_0%X020(Gyou, Retu)
Rem PRVLG
  CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P
    = CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W%X300_0%X290_INC_FLG True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%X340 < 0%X350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%X340 > 0%X350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%X340 < 0%X350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If

```

## [図117]

定義規則 10 で定義されるベクトル I.4 (正規) - 4

```
Rem PKVLG
  If @%34@ > @%35@ then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PRCEND
```



## [図118]

定義規則11で定義されるベクトルL3 (K) - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB4N3
Private W%X300_0%X310_0%X280_0%X290_0%X020 As Integer
Rem PUBIN3
Private W%X3050_0%X3150_0%X2850_0%X2950_0%X2000 As 0%X2350
Rem PUB5C3
Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P As Integer
Rem PUB5D3
Private CTRL_W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEF PRV
Rem PRV2T3
Private W%X300_0%X310_0%X280_0%X290_0%X020 wk As Integer
Rem PRV7R3
Private CTRL_W%X300_0%X290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%X1620
Rem $PRV2E3
Rem $PRV4W3
0%X1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%X310_0%X280_0%X290_0%X020()
BOX_1:
Rem PRVLG
If W%X300_0%X310_0%X280_0%X290_0%X020 = CNS_NOT_KUH_Integer Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%X160
BOX_3:
Rem PRVLG
If W%X300_0%X310_0%X280_0%X290_0%X020_wk <> CNS_NOT_KUH_Integer Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG

```

[図119]

定義規則1 1で定義されるベクトルL 3 (K) - 2

```

W0X300_0X310_0X280_0X290_0X020 = W0X300_0X310_0X280_0X290_0X020_wk
Rem PRVLG
  CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG = CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG_P
    = CTRL_W0X300_0X280_0X290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0X300_0X310_0X280_0X290_0X020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0X300_0X290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

## [図120]

定義規則 1 1 で定義されるベクトル L 4 (K) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020_Sum As 0%230
Rem PUB4N4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5V4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5R4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBK11
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG

```

[図121]

定義規則 1 1 で定義されるベクトル L 4 (K) - 2

```

      If W0%300_0%310_0%280_0%290_0%020_Sum = CNS_NOT_KUH_0%230 And _
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%3050_0%3150_0%2850_0%2950_0%2000 - CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
      0%170
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020_Sum = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      W2_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      CTRL W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

[図122]

定義規則 1 1 で定義されるベクトル L 2 (K) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N2
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB1N2
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F2
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T2
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R2
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUHL_0%230 As 0%230
Rem $PRV2W2
0%1520
Rem $PRV2E2
Rem $PRV4W2
0%1550
Rem $PRV4E2
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L2_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUHL_0%230 Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
' NOP
BOX_3:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUHL_0%230 Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG
W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020

```

## [図123]

定義規則 1.1 で定義されるベクトル L.2 (K) - 2

```
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND
```



## [図124]

定義規則 1 1 で定義されるベクトル L 3 (K の派生元の正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
  Private W0%300_0%310_0%280_0%290_0%020 As Integer
Rem PUBIN3
  Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6P3
  Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
  Private W0%300_0%310_0%280_0%290_0%020_wk As Integer
Rem PRV7R3
  Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
  BOX_1:
Rem PRVLG
    If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_Integer Then
      GoTo BOX_2
    End If
    GoTo BOX_E
  BOX_2:
Rem PRVLG
    0%160
  BOX_3:
Rem PRVLG
    If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_Integer Then
      GoTo BOX_4
    End If
    GoTo BOX_5
  BOX_4:
Rem PRVLG

```

## [図125]

定義規則 1.1 で定義されるベクトル L 3 (Kの派生元の正規) - 2

```

70%300_0%310_0%280_0%290_0%020 = 70%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
  CTRL_70%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_70%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_70%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_70%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_70%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_70%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```

## [図126]

定義規則 11 で定義されるバクトル L4 (K の派生元の正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB1N4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEPEND
Rem PUB4D4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E1
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG

```

## [図127]

定義規則 1 1 で定義されるベクトル L 4 (K の派生元の正規) - 2

```

      If W0%300_0%310_0%280_0%290_0%020_wk = CNS_NOT_KUH_0%230 And _
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%3050_0%3150_0%2850_0%2950_0%2000 = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020_wk = W0%3050_0%3150_0%2850_0%2950_0%2000
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_INC_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

[図128]

定義規則12で定義されるベクトルL3 (K) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB1N3
Private W0%300_0%310_0%280_0%290_0%020 As Integer
Rem PUB1N3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_0%020_wk0%070 As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH Integer Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%1600
BOX_3:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then GoTo BOX_4
GoTo BOX_5
BOX_4:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_0%020_wk0%070 = 1 Then W0%300_0%310_0%280_0%290_0%020 = 0%070
Rem PRVLG

```

[図129]

定義規則 1 2 で定義されるベクトル L 3 (K) - 2

```

      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG - 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
        - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
          GoTo BOX_6
        End If
          GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_LNG_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```



## [図130]

定義規則 1 2 で定義されるベクトル L 4 (K) - 1

```

Option Explicit
Rem #DEFPU3
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020_Sum As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000 As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020 As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_0%0 As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820
Rem $PRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem PRVLC

```

## [図131]

定義規則 1 2 で定義されるベクトル 1.4 (K) - 2

```

      If W0%300_0%310_0%280_0%290_0%020_Sum = CNS_NOT_KUH_0%230 And _
Rem PRVLG
      W3_0%310_0%280_0%290_0%020 = 0%070 Then
      GoTo BOX_2
      End If
      GoTo BOX_E
BOX_2:
Rem PUBIN
      If W0%300_0%310_0%280_0%290_0%020 = CNS_NOT_KUH_0%235 Then Go To Box_3
Rem PRVLG
      0%170
BOX_3:
Rem PRVLG
      If W0%300_0%310_0%280_0%290_0%020_wk <> CNS_NOT_KUH_0%230 Then
      GoTo BOX_4
      End If
      GoTo BOX_5
BOX_4:
Rem PRVLG
      W0%300_0%310_0%280_0%290_0%020_Sum = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      W2_0%310_0%280_0%290_0%020 = W0%300_0%310_0%280_0%290_0%020_wk
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_INC_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

## [図132]

定義規則13で定義されるベクトルL3(K) - 1

```

Option Explicit
Rem #DEF PUB
Rem PUB4N3
Private W%300_0%310_0%280_0%290_0%020(0%260, 0%270) As Integer
Rem PUBIN3
Private W%3050_0%3150_0%2850_0%2950_0%2000(0%2650, 0%2750) As 0%2350
Rem PUB5C3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260, 0%270) As Boolean
Rem #DEFEND
Rem #DEF PRV
Rem PRV2T3
Private W%300_0%310_0%280_0%290_0%020_wk(0%260, 0%270) As Integer
Rem PRV7R3
Private CTRL_W%300_0%290_ING_FLG As Boolean
Rem PUBKL
Private CNS_NOT_KUH_Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then

```

## [図133]

定義規則 13 で定義されるベクトル L3 (K) - 2

```

Rem PRVLG
    Uti_Max = 0%270
Rem PRVLG
    Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max - 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0%160
BOX_3:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS_NOT_KUH_0%230 Then
            GoTo BOX_4
        End If
        GoTo BOX_5
BOX_4:
Rem PRVLG
        W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
        CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%260_0%290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        IF CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
            = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7

```

## [図134]

定義規則 13 で定義されるベクトル L3 (K) - 3

```

BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%290_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End if
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt - 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PROCEND

```

## [図135]

定義規則 13 で定義されるベクトル L4 (K) - 1

```

Option Explicit
Rem #DEFPCB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020_Sun(0%260, 0%270) As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650, 0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260, 0%270) As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260, 0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260, 0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260, 0%270) As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260, 0%270) As 0%230
Rem PRV7K4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem #PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```



## [図136]

定義規則13で定義されるベクトルL4 (K) - 2

```

Rem $PRV4E4
Rem $DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
        End If
Rem PRVLG
    If 0%340 > 0%350 then
Rem PRVLG
        Uti_Max = 0%270
Rem PRVLG
        Soto_Max = 0%280
        End If
        Uti_Cnt = 0
        Soto_Cnt = 0
        Do Until Soto_Cnt = Soto_Max + 1
            Uti_Cnt = 0
Rem PRVLG
            If 0%340 < 0%350 then
                Gyou = Uti_Cnt
                Retu = Soto_Cnt
            End If
Rem PRVLG
            If 0%340 > 0%350 then
                Gyou = Soto_Cnt
                Retu = Uti_Cnt
            End If
            Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
            If 0%300_0%310_0%280_0%290_0%020_Sum(Gyou, Retu) = CNS_NOT_KUH_0%230 And _
Rem PRVLG
                0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
                    GoTo BOX_2
            End If
            GoTo BOX_F
BOX_2:
Rem PUBIN
            If 0%3050_0%3150_0%2850_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
            0%170

```

## [図137]

定義規則 13 で定義されるベクトル L 4 (K) - 3

```

BOX_3:
Rem PRVLG
  If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS_NOT_KUH_0%230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W0%300_0%310_0%280_0%290_0%020_Sum(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
  W2_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If

```

## [図138]

定義規則 13 で定義されるベクトル L 4 (K) - 4

```
Rec PRVLG
  If @%34@ > @%35@ then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PRCEND
```

## [図139]

定義規則 13 で定義されるベクトル L2 (K) - 1

```

Option Explicit:
Rem #DEFPUB
Rem PUB4N2
Private W%300_0%310_0%280_0%290_0%020(0%260, 0%270) As 0%230
Rem PUBIN2
Private W%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
Private CTRL_W%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F2
Private CTRL_W%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260, 0%270) As Boolean
Rem #DEPEND
Rem #DEFPRV
Rem PRV2T2
Private W%300_0%310_0%280_0%290_0%020_wk(0%260, 0%270) As 0%230
Rem PRV7R2
Private CTRL_W%300_0%290_ING_FLG As Boolean
Rem PURKU
Private CNS_NOT_KUIL_0%230 As 0%230
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem SPRV2W2
0%1520
Rem $PRV2E2
Rem $PRV4W2
0%1550
Rem $PRV4E2
Rem #DEPEND
Public Sub Main()
Rem #VECREP Private Sub L2_0%310_0%230_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLC
    If 0%340 > 0%350 then

```

## [図140]

定義規則 13 で定義されるベクトル L 2 (K) - 2

```

Rem PRVLG
    Uti_Max = 0x270
Rem PRVLG
    Soto_Max = 0x260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0x340 < 0x350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0x340 > 0x350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = CNS_NOT_KUH_0x230 Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
        ' NOP
BOX_3:
Rem PRVLG
        If W0x300_0x310_0x280_0x290_0x020_wk(Gyou, Retu) <> CNS_NOT_KUH_0x230 Then
            GoTo BOX_4
        End If
        GoTo BOX_5
BOX_4:
Rem PRVLG
        W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = W0x300_0x310_0x280_0x290_0x020(Gyou, Retu)
Rem PRVLG
        CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        IF CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_P
            = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7
BOX_6:

```

[図141]

定義規則 13 で定義されるベクトル L 2 (K) - 3

```

Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG (Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt - 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt - 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PRCEND

```



## [図142]

定義規則 13 で定義されるベクトル L 3 (K の派生元の正規) - 1

```

Option Explicit
Rem #DEFPOB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_0%020(0%260, 0%270) As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650, 0%2750) As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260, 0%270) As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_0%020_Wk(0%260, 0%270) As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%1820
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
        End If
Rem PRVLG
    If 0%340 > 0%350 then

```

## [図143]

定義規則 13 で定義されるベクトル L 3 (K の派生元の正規) - 2

```

Rem PRVLG
    Uti_Max = 0x270
Rem PRVLG
    Soto_Max = 0x260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0x340 < 0x350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt    End If
Rem PRVLG
        If 0x340 > 0x350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt    End If
        Do Until Uti_Cnt = Uti_Max + 1
            BOX_1:
            Rem PRVLG
                If W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
                    GoTo BOX_2
                End If
                GoTo BOX_E
            BOX_2:
            Rem PRVLG
                0x160
            BOX_3:
            Rem PRVLG
                If W0x300_0x310_0x280_0x290_0x020_wk(Gyou, Retu) <> CNS_NOT_KUH_Integer
                    Then GoTo BOX_4
                End If
                GoTo BOX_5
            BOX_4:
            Rem PRVLG
                W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = W0x300_0x310_0x280_0x290_0x020_wk(Gyou, Retu)
            Rem PRVLG
                CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG + 1
                GoTo BOX_E
            BOX_5:
            Rem PRVLG
                If CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_P
                    = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_PP Then GoTo BOX_6 End If GoTo BOX_7
            BOX_6:
            Rem PRVLG
                CTRL_W0x300_0x310_0x280_0x290_0x020_NOT_VALID_FLG(Gyou, Retu) = True
                GoTo BOX_E
            BOX_7:

```

## [図144]

定義規則 13 で定義されるベクトル L 3 (K の派生元の正規) - 3

```

Rem PRVLG
  CTRL_W@%300_@%290_ING_FLG = True GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If @%340 < @%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt End If
Rem PRVLG
  If @%340 > @%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If @%340 < @%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt End If
Rem PRVLG
  If @%340 > @%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PROCEND

```

[図145]

定義規則 13 で定義されるベクトル L 4 (K の派生元の正規) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%230_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem PUB4B4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260,0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260,0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260,0%270) As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```

## [図146]

定義規則 1 3 で定義されるベクトル L 4 (Kの派生元の正規) - 2

```

Rem $PRV1E1
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L4_0%310_0%280_0%290_0%020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
    End If
Rem PRVLG
    If 0%340 > 0%350 then
Rem PRVLG
        Uti_Max = 0%270
Rem PRVLG
        Soto_Max = 0%260
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0%340 < 0%350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0%340 > 0%350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX 1:
Rem PRVLG
        If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) = CNS_NOT_KUH_0%230 And _
Rem PRVLG
        W3_0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PUBIN
        If W0%3050_0%3150_0%2850_0%2950_0%2000(Gyou, Retu) = CNS_NOT_KUH_0%2350 Then Go To Box_3
Rem PRVLG
        W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) = W0%3050_0%3150_0%2850_0%2950_0%2000(Gyou, Retu)

```

## [図147]

定義規則 13 で定義されるベクトル L 4 (K の派生元の正規) - 3

```

BOX_3:
Rem PRVLG
  If W0%300_0%310_0%280_0%290_0%020_vk(Gyou, Retu) <> CNS_NOT_KUH_0%230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_vk(Gyou, Retu)
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG - 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_INC_FLG = True
  GoTo BOX_E
BOX E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then

```



## [図148]

定義規則 13 で定義されるベクトル L 4 (K の派生元の正規) - 4

```
Gyou = Soto_Cnt  
Retu = Uti_Cnt  
End If  
Loop  
Gyou = 0  
Retu = 0  
Uti_Cnt = 0  
Soto_Cnt = 0  
Uti_Max = 0  
Soto_Max = 0  
End Sub  
Rem *** PRCEND
```

[図149]

定義規則14で定義されるベクトルL3 (K) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
Private W%X300_0%X310_0%X280_0%X290_0%X020(0%X260,0%X270) As Integer
Rem PUB1N3
Private W%X3050_0%X3150_0%X2850_0%X2950_0%X2000(0%X2650,0%X2750) As 0%X2350
Rem PUB5C3
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL W%X300_0%X280_0%X290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL W%X300_0%X310_0%X280_0%X290_0%X020_NOT_VALID_FLG(0%X260,0%X270) As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W%X300_0%X310_0%X280_0%X290_0%X020_Wk0%X070(0%X260,0%X270) As Integer
Rem PRV7R3
Private CTRL W%X300_0%X290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%X1620
Rem $PRV2E3
Rem $PRV4W3
0%X1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3 0%X310_0%X280_0%X290_0%X020()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%X340 < 0%X350 then
Rem PRVLG
        Uti_Max = 0%X260
Rem PRVLG
        Soto_Max = 0%X270
    End If
Rem PRVLG
    If 0%X340 > 0%X350 then

```

## [図150]

定義規則 14 で定義されるベクトル L 3 (K) - 2

```

Rem PRVLG
    Uti_Max = 0x270
Rem PRVLG
    Soto_Max = 0x250
    End If
    Uti_Cnt = 0
    Soto_Cnt = 0
    Do Until Soto_Cnt = Soto_Max + 1
        Uti_Cnt = 0
Rem PRVLG
        If 0x340 < 0x350 then
            Gyou = Uti_Cnt
            Retu = Soto_Cnt
        End If
Rem PRVLG
        If 0x340 > 0x350 then
            Gyou = Soto_Cnt
            Retu = Uti_Cnt
        End If
        Do Until Uti_Cnt = Uti_Max + 1
BOX_1:
Rem PRVLG
        If W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = CNS_NOT_KUH_Integer Then
            GoTo BOX_2
        End If
        GoTo BOX_E
BOX_2:
Rem PRVLG
        0x160
BOX_3:
Rem PUBL3F
        If W0x300_0x310_0x280_0x290_0x020_0x070(Gyou, Retu) = 1 Then GoTo BOX_4
        GoTo BOX_5
BOX_4:
Rem PUBL3F
        If W0x300_0x310_0x280_0x290_0x020_0x070(Gyou, Retu) = 1
            Then W0x300_0x310_0x280_0x290_0x020(Gyou, Retu) = 0x070
Rem PRVLG
        CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG + 1
        GoTo BOX_E
BOX_5:
Rem PRVLG
        If CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_P
            = CTRL_W0x300_0x280_0x290_STS_TRANSITION_FLG_PP Then
            GoTo BOX_6
        End If
        GoTo BOX_7
BOX_6:

```

[図151]

定義規則 14 で定義されるベクトル L 3 (K) - 3

```

Rem PRVLG
  CTRL_#0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_#0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Gyou = 0
  Retu = 0
  Uti_Cnt = 0
  Soto_Cnt = 0
  Uti_Max = 0
  Soto_Max = 0
End Sub
Rem *** PRCEND

```

## [図152]

定義規則14で定義されるベクトルL4(K) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020_Sum(0%260,0%270) As 0%230
Rem PUBIN4
Private W0%3050_0%3150_0%2850_0%2950_0%2000(0%2650,0%2750) As 0%2350
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(0%260,0%270) As Boolean
Rem #DEFEND
Rem PUB4D4
Private W0%3000_0%3100_0%2800_0%2900_0%2000(0%260,0%270) As 0%230
Rem PUB4N3
Private W3_0%310_0%280_0%290_0%020(0%260,0%270) As Integer
Rem PUB4N2
Private W2_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Private Gyou as Integer
Private Rotu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem #DEFPRV
Rem PRV2T4
Private W0%300_0%310_0%280_0%290_0%020_wk(0%260,0%270) As 0%230
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem *PUBKU
Private CNS_NOT_KUH_String As String
Private CNS_NOT_KUH_Integer As Integer
Private CNS_NOT_KUH_Boolean As Boolean
Private CNS_NOT_KUH_Long As Long
Private CNS_NOT_KUH_Single As Single
Private CNS_NOT_KUH_Double As Double
Private CNS_NOT_KUH_Variant As Variant
Private CNS_NOT_KUH_Currency As Currency
Private CNS_NOT_KUH_Byte As Byte
Private CNS_NOT_KUH_Date As Date
Rem $PRV2W4
0%1720
Rem $PRV2E4
Rem $PRV4W4
0%1820

```

[図153]

定義規則 14 で定義されるベクトル L4 (K) - 2

```

Rem $PRV4E4
Rem hDEPEND
Public Sub Main()
Rem #VECRPP Private Sub L4_0%310_0%280_0%290_0%020_0%070()
    Gyou = 0
    Retu = 0
Rem PRVLG
    If 0%340 < 0%350 then
Rem PRVLG
        Uti_Max = 0%260
Rem PRVLG
        Soto_Max = 0%270
        End If
Rem PRVLG
        If 0%340 > 0%350 then
Rem PRVLG
            Uti_Max = 0%270
Rem PRVLG
            Soto_Max = 0%260
            End If
            Uti_Cnt = 0
            Soto_Cnt = 0
            Do Until Soto_Cnt = Soto_Max + 1
                Uti_Cnt = 0
Rem PRVLG
                If 0%340 < 0%350 then
                    Gyou = Uti_Cnt
                    Retu = Soto_Cnt
                End If
Rem PRVLG
                If 0%340 > 0%350 then
                    Gyou = Soto_Cnt
                    Retu = Uti_Cnt
                End If
                Do Until Uti_Cnt = Uti_Max - 1
BOX_1:
Rem PRVLG
                If W0%300_0%310_0%280_0%290_0%020_Sum(Gyou, Retu) = CNS_NOT_KUH_0%230 And _
Rem PRVLG
                WJ_0%310_0%280_0%290_0%020(Gyou, Retu) = 0%070 Then
                    GoTo BOX_2
                End If
                GoTo BOX_E
BOX_2:
Rem PUBIN
                If W0%300_0%310_0%280_0%290_0%020(Gyou, Retu) = CNS_NOT_KUH_0%235 Then Go To Box_3
Rem PRVLG
                0%170

```

## [図154]

定義規則 1 4 で定義されるベクトル L 4 (K) - 3

```

BOX_3:
Rem PRVLG
  If W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu) <> CNS_NOT_KUH_0%230 Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  W0%300_0%310_0%280_0%290_0%020_Sum(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
  W2_0%310_0%280_0%290_0%020(Gyou, Retu) = W0%300_0%310_0%280_0%290_0%020_wk(Gyou, Retu)
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG(Gyou, Retu) = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
  Uti_Cnt = Uti_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
  Loop
  Soto_Cnt = Soto_Cnt + 1
Rem PRVLG
  If 0%340 < 0%350 then
    Gyou = Uti_Cnt
    Retu = Soto_Cnt
  End If

```



## [図155]

定義規則 14 で定義されるベクトル L4 (K) - 4

```
Rem PRVLG
  If 0%340 > 0%350 then
    Gyou = Soto_Cnt
    Retu = Uti_Cnt
  End If
Loop
Gyou = 0
Retu = 0
Uti_Cnt = 0
Soto_Cnt = 0
Uti_Max = 0
Soto_Max = 0
End Sub
Rem *** PCEND
```

[図156]

定義規則 19 で定義されるベクトル 12 (入力論理体) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N2
Private CTR_0%310_0%280_0%290_I2_END As Boolean
Rem PUB4N2
Private CTR_0%310_0%280_0%290_I2_CNT As Integer
Rem PUB4N2
Private CTR_0%310_0%280_0%290_I2_ERR As Boolean
Rem PUB4N2
Private CTR_0%310_0%280_0%290_I2_EOF As Boolean
Rem PUBIN2
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem I2AKYARA
Rem I2PKYARA
Rem PUB6F2
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R2
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem $PRV2W2
0%1520
Rem $PRV2E2
Rem $PRV4W2
0%1550
Rem $PRV4E2
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub I2_0%310_0%280_0%290_0%020_0%070()
BOX_1:
Rem I2PKYKUU
If 0%500 <> 0%070 Then Exit Sub
Rem I2AKYKUU
If 0%500 <> 0%070 Then Exit Sub
BOX_2:
Rem PRVLC
0%160
Rem PRVLC
CTR_0%310_0%280_0%290_I2_END = False
Rem PRVLC
CTR_0%310_0%280_0%290_I2_ERR = False
BOX_3:

```

## [図157]

定義規則 19 で定義されるベクトル I 2 (入力論理体) - 2

```

Rem PRVLG
  If CTR_0%310_0%280_0%290_I2_END = False Then
    GoTo BOX_4
  End If
  GoTo BOX_5
BOX_4:
Rem PRVLG
  0%1530
Rem PRVLG
  CTR_0%310_0%280_0%290_I2_END = True
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTR_0%310_0%280_0%290_I2_END = False Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTR_0%310_0%280_0%290_I2_ERR = True
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

[図158]

定義規則 20 で定義されるベクトル L 3 (入力アクセスキー) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_IN_ACCESS_KEY As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_IN_ACCESS_KEY_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_IN_ACCESS_KEY wk0%070 As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH Integer As Integer
Private Gyou as Integer
Private Retu as Integer
Private Uti_Cnt as Integer
Private Soto_Cnt as Integer
Private Uti_Max as Integer
Private Soto_Max as Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_IN_ACCESS_KEY()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_IN_ACCESS_KEY = W0%300_0%310_0%280_0%290_IN_ACCESS_KEY Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PUBL3F

```

[図159]

定義規則 20 で定義されるベクトル L 3 (入力アクセスキー) - 2

```

      If W0%300_0%310_0%280_0%290_IN_ACCESS_KEY_wk0%070 = 1 Then GoTo BOX_4
      GoTo BOX_5
BOX_4:
Rem PUBL3F
      If W0%300_0%310_0%280_0%290_IN_ACCESS_KEY_wk0%070 = 1
      Then W0%300_0%310_0%280_0%290_IN_ACCESS_KEY = 0%070
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_IN_ACCESS_KEY_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

[図160]

定義規則 2 1 で定義されるベクトル L 3 (入力処理条件キー) - 1

```

Option Explicit
Rem #DEFIUB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_IN_Conditions_KEY As Integer
Rem PUB1N3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_IN_Conditions_KEY_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_IN_Conditions_KEY_wk0%070 As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_IN_Conditions_KEY()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_IN_Conditions_KEY = W0%300_0%310_0%280_0%290_IN_Conditions_KEY
Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_IN_Conditions_KEY_wk0%070 = 1 Then GoTo BOX_4
GoTo BOX_5
BOX_4:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_IN_Conditions_KEY_wk0%070 = 1

```

## [図161]

定義規則 2 1 で定義されるベクトル L 3 (入力処理条件キー) - 2

```

      Then W0%300_0%310_0%250_0%290_IN_Conditions_KEY = 0%070
Rem PRVLG
      CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
      = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W0%300_0%310_0%280_0%290_IN_Conditions_KEY_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W0%300_0%290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```



## [図162]

定義規則 2 2 で定義されるベクトル 0 4 (出力論理体) - 1

```

Option Explicit
Rem #DEFMUB
Public CTR_NEXT_PALLET_ID As String
Type RouteControlTableType
    No As Integer
    SFID As String
    Row As Integer
    Col As Integer
    PalID As String
    SubNo As Integer
    RouteCode As String
    NextPallets As Variant
    NPID_R0R As String
    NPID_R3M As String
    NPID_R3D As String
    NPID_R3C As String
    O4_ID As String
    SendedFLG As String
End Type
Public RctIndex1 As Integer
Public RctIndex2 As Integer
Public RctIndex3 As Integer
Public RCI(10000) As RouteControlTableType
Rem PUB4N4
    Private CTR_0%310_0%280_0%290_O4_END As Boolean
Rem PUB4N4
    Private CTR_0%310_0%280_0%290_O4_CNT As Integer
Rem PUB4N4
    Private CTR_0%310_0%280_0%290_O4_ERR As Boolean
Rem PUBIN4
    Private 0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C4
    Private CTRL_0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
    Private CTRL_0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
    Private CTRL_0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem O4AKYARA
Rem O4PKYARA
Rem PUB6F4
    Private CTRL_0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
    Private CTRL_0%300_0%290_ING_FLG As Boolean
Rem $PRV2W4
0%1720
Rem $PRV2E4

```

## [図163]

定義規則 2 2 で定義されるベクトル O 4 (出力論理体) - 2

```

Rem $PRV4W4
@%1828
Rem $PRV4E4
Rem #DEFEND
Rem *-----
Public Sub Main()
Rem #VECREP Private Sub O4_@%318_@%288_@%298_@%028_@%078()
BOX_1:
Rem O4AKYKUU
If @%508 <> @%078 Then Exit Sub
Rem O4PKYKUU
If @%508 <> @%078 Then Exit Sub
BOX_2:
Rem PRVLG
@%178
Rem PRVLG
CTR_@%318_@%288_@%298_O4_END = False
Rem PRVLG
CTR_@%318_@%288_@%298_O4_ERR = False
BOX_3:
Rem PRVLG
If CTR_@%318_@%288_@%298_O4_END = False Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG
@%1538
Rem PRVLG
CTR_@%318_@%288_@%298_O4_END = True
Rem PRVLG
CTRL_W@%308_@%288_@%298_STS_TRANSITION_FLG = CTRL_W@%308_@%288_@%298_STS_TRANSITION_FLG + 1
RctIndex1 = 0
Do Until RCT(RctIndex1).No < 1
If RCT(RctIndex1).NPID_R3R = CTR_NEXT_PALETTE_ID Then
RctIndex2 = 0
Do Until RCT(RctIndex2).No < 1
If RCT(RctIndex2).SFID = RCT(RctIndex1).SFID _
Rem PRVLG
And RCT(RctIndex2).O4_ID = "O4_@%318_@%288_@%298_@%028_@%078" Then
And RCT(RctIndex2).SendedFLG = "1"
Exit Do
End If
RctIndex2 = RctIndex2 + 1
Loop
End If
RctIndex1 = RctIndex1 + 1
Loop

```

## [図164]

定義規則 2 2 で定義されるベクトル O 4 (出力論理体) - 3

```
        GoTo BOX_E
BOX_5:
Rem PRVLG
    If CTR_0%310_0%280_0%290_04_END = False Then
        GoTo BOX_6
    End If
    GoTo BOX_7
BOX_6:
Rem PRVLG
    CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
Rem PRVLG
    CTR_0%310_0%280_0%290_04_ERR = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W0%300_0%290_ING_FLG = True
    GoTo BOX_E
BOX_E:
End Sub
Rem *** PCEND
```

## [図165]

定義規則 2 3 で定義されるベクトル L 3 (出力アクセスキー) - 1

```

Option Explicit:
Rem #DEFPUB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY As Integer
Rem PUBJN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2350
Rem PUB5C3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL_W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY_wk0%070 As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_OUT_ACCESS_KEY()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY = W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY_wk0%070 = 1 Then GoTo BOX_4
GoTo BOX_5
BOX_4:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY_wk0%070 = 1
Then W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY = 0%070

```

## [図166]

定義規則 2 3 で定義されるベクトル L 3 (出力アクセスキー) - 2

```
Rem PRVLG
  CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG = CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG + 1
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P
    - CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG PP Then
    GoTo BOX_5
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_OUT_ACCESS_KEY_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND
```

## [図167]

定義規則 2 4 で定義されるベクトル L 3 (出力処理条件キー) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB4N3
Private W0%300_0%310_0%280_0%290_OUT_Conditions_KEY As Integer
Rem PUBIN3
Private W0%3050_0%3150_0%2850_0%2950_0%2000 As 0%2950
Rem PUB5C3
Private CTRL W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F3
Private CTRL W0%300_0%310_0%280_0%290_OUT_Conditions_KEY_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEPPRV
Rem PRV2T3
Private W0%300_0%310_0%280_0%290_OUT_Conditions_KEY_wk0%070 As Integer
Rem PRV7R3
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub L3_0%310_0%280_0%290_OUT_Conditions_KEY()
BOX_1:
Rem PRVLG
If W0%300_0%310_0%280_0%290_OUT_Conditions_KEY - W0%300_0%310_0%280_0%290_OUT_Conditions_KEY
Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
0%160
BOX_3:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_OUT_Conditions_KEY_wk0%070 = 1 Then GoTo BOX_4
GoTo BOX_5
BOX_4:
Rem PUBL3F
If W0%300_0%310_0%280_0%290_OUT_Conditions_KEY_wk0%070 = 1

```

## [図168]

定義規則 2 4 で定義されるベクトル L 3 (出力処理条件キー) - 2

```

      Then W%X300_0X310_0X280_0X290_OUT_Conditions_KEY = 0X070
Rem PRVLG
      CTRL_W%X300_0X280_0X290_STS_TRANSITION_FLG = CTRL_W%X300_0X280_0X290_STS_TRANSITION_FLG - 1
      GoTo BOX_E
BOX_5:
Rem PRVLG
      If CTRL_W%X300_0X280_0X290_STS_TRANSITION_FLG_P
      = CTRL_W%X300_0X280_0X290_STS_TRANSITION_FLG_PP Then
      GoTo BOX_6
      End If
      GoTo BOX_7
BOX_6:
Rem PRVLG
      CTRL_W%X300_0X310_0X280_0X290_OUT_Conditions_KEY_NOT_VALID_FLG = True
      GoTo BOX_E
BOX_7:
Rem PRVLG
      CTRL_W%X300_0X290_ING_FLG = True
      GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND

```



## [図169]

定義規則 25 で定義されるベクトル R 4 (WO 4 パレット) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB5C4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B4
Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F4
Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem *PUBRT
Private CTR_NEXT_PALLET_ID_NEW As String
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_String As String
Rem SPRV2W4
0%1720
Rem SPRV2E4
Rem SPRV4W4
0%1820
Rem SPRV4E4
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub R4_0%310_0%280_0%290_0%020()
BOX_1:
Rem PRVLG
If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
CTR_NEXT_PALLET_ID_NEW = "0%1000"
BOX_3:
Rem PRVLG
If CTR_NEXT_PALLET_ID_NEW <> CNS_NOT_KUH_String Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG
CTR_NEXT_PALLET_ID_NEW = CTR_NEXT_PALLET_ID_NEW
GoTo BOX_E
BOX_5:
Rem PRVLG

```

## [図170]

定義規則 25 で定義されるベクトル R 4 (W 0 4 パレット) - 2

```
    If CTR_NEXT_PALLET_ID_NE# = CNS_NOT_KUH_String Then
        GoTo BOX_6
    End If
    GoTo BOX_7
BOX_6:
Rem PRVLG
    CTRL_W0%300_0%310_0%290_0%290_0%020_NOT_VALID_FLG = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W0%300_0%290_ING_FLG = True
    GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND
```

## [図171]

定義規則 2.6 で定義されるベクトル R 2 C (W 0 2 パレット) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB5C2
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B2
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F2
  Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem *PUBRT
  Private CTR_NEXT_PALLET_ID_NEW As String
Rem PRV7R2
  Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_String As String
Rem SPRV2W2
0%1520
Rem SPRV2E2
Rem SPRV4W2
0%1550
Rem SPRV4E2
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub R2C_0%310_0%280_0%290_0%020()
  BOX_1:
Rem PRVLG
    If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
      GoTo BOX_2
    End If
    GoTo BOX_E
  BOX_2:
Rem PRVLG
    0%150
Rem PRVLG
    CTR_NEXT_PALLET_ID_NEW = "0%1000"
  BOX_3:
Rem PRVLG
    If CTR_NEXT_PALLET_ID_NEW <> CNS_NOT_KUH_String Then
      GoTo BOX_4
    End If
    GoTo BOX_5
  BOX_4:
Rem PRVLG
    0%1630
Rem PRVLG

```

## [図172]

定義規則26で定義されるベクトルR2C (WO2パレット) - 2

```
CTR_NEXT_PALLET_ID_NEW = CTR_NEXT_PALLET_ID_NEW
  GoTo BOX_E
BOX_5:
Rem PRVLG
  If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
    GoTo BOX_6
  End If
  GoTo BOX_7
BOX_6:
Rem PRVLG
  CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
  GoTo BOX_E
BOX_7:
Rem PRVLG
  CTRL_W0%300_0%290_ING_FLG = True
  GoTo BOX_E
BOX_E:
End Sub
Rem *** PROCEND
```

[図173]

定義規則 27 で定義されるベクトル R 2 (W O 2 パレット) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB5C2
Private CTRL_W@%30@_@%28@_@%29@_STS_TRANSITION_FLG As Integer
Rem PUB5Y2
Private CTRL_W@%30@_@%28@_@%29@_STS_TRANSITION_FLG_P As Integer
Rem PUB5D2
Private CTRL_W@%30@_@%28@_@%29@_STS_TRANSITION_FLG_PP As Integer
Rem PUB6F2
Private CTRL_W@%30@_@%31@_@%28@_@%29@_@%02@_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem *PUBRT
Private CTR_NEXT_PALLET_ID_NEW As String
Rem PRV7R2
Private CTRL_W@%30@_@%29@_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_String As String
Rem $PRV2W2
@%152@
Rem $PRV2E2
Rem $PRV4W2
@%155@
Rem $PRV4E2
Rem #DBFEND
Public Sub Main()
Rem #VECREP Private Sub R2_@%31@_@%28@_@%29@_@%02@_()
BOX_1:
Rem PRVLG
If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
GoTo BOX_2
End If
GoTo BOX_E
BOX_2:
Rem PRVLG
CTR_NEXT_PALLET_ID_NEW = "@%100@"
BOX_3:
Rem PRVLG
If CTR_NEXT_PALLET_ID_NEW <> CNS_NOT_KUH_String Then
GoTo BOX_4
End If
GoTo BOX_5
BOX_4:
Rem PRVLG
CTR_NEXT_PALLET_ID_NEW = CTR_NEXT_PALLET_ID_NEW
GoTo BOX_E
BOX_5:
Rem PRVLG

```

## [図174]

定義規則 27 で定義されるベクトル R 2 (W O 2 パレット) - 2

```
If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
    GoTo BOX_6
End If
    GoTo BOX_7
BOX_6:
Rem PRVLG
    CTRL_W0%300_0%310_0%250_0%290_0%020_NOT_VALID_FLG = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W0%300_0%290_ING_FLG = True
    GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND
```

[図175]

定義規則 28 で定義されるベクトル R3R (WO 3 パンット) - 1

```

Option Explicit
Rem #DEFPUB
Rem PUB5C3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5Y3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_P As Integer
Rem PUB5B3
  Private CTRL_W0%300_0%280_0%290_STS_TRANSITION_FLG_PP As Integer
Rem PUB5C2
  Private CTRL_W2_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB5C4
  Private CTRL_W4_0%280_0%290_STS_TRANSITION_FLG As Integer
Rem PUB6F3
  Private CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem *PUBRT
  Private CTR_NEXT_PALLET_ID_NEW As String
Rem PRV7R3
  Private CTRL_W0%300_0%290_INC_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_String As String
Rem $PRV2W3
0%1620
Rem $PRV2E3
Rem $PRV4W3
0%1650
Rem $PRV4E3
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub R3R_0%310_0%280_0%290_0%020()
  BOX_1:
Rem PRVLG
    If CTR_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String And _
Rem PRVLG
      (CTRL_W4_0%280_0%290_STS_TRANSITION_FLG <> 0 Or _
Rem PRVLG
      CTRL_W2_0%280_0%290_STS_TRANSITION_FLG <> 0) Then
      GoTo BOX_2
    End If
    GoTo BOX_E
  BOX_2:
Rem PRVLG
    CTR_NEXT_PALLET_ID_NEW = "0%1000"
  BOX_3:
Rem PRVLG
    If CTR_NEXT_PALLET_ID_NEW <> CNS_NOT_KUH_String Then
      GoTo BOX_4

```



## [図176]

定義規則 28 で定義されるベクトル R 3 R (W 0 3 パレット) - 2

```

    End If
    GoTo BOX_5
BOX_4:
Rem PRVLG
    CTRL_NEXT_PALLET_ID_NEW = CTRL_NEXT_PALLET_ID_NEW
    GoTo BOX_E
BOX_5:
Rem PRVLG
    If CTRL_NEXT_PALLET_ID_NEW = CNS_NOT_KUH_String Then
        GoTo BOX_6
    End If
    GoTo BOX_7
BOX_6:
Rem PRVLG
    CTRL_W0%300_0%310_0%280_0%290_0%020_NOT_VALID_FLG = True
    GoTo BOX_E
BOX_7:
Rem PRVLG
    CTRL_W0%300_0%290_LNG_FLG = True
    GoTo BOX_E
BOX_E:
End Sub
Rem *** PRCEND

```

## [図177]

定義規則 33 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4M4
    Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
    Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
    Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04013()
    BOX_1:
Rem PUBIN
    BOX_2:
Rem PUBKK
    BOX_3:
    BOX_4:
    BOX_5:
    BOX_6:
    BOX_7:
    BOX_E:
End Sub
Rem *PRCEND

```

## [図178]

定義規則34で定義されるベクトルS4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
  Private W%X30%-X%31%-X%28%-X%29%-X%02% As X%23%
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
  Private CNS_NOT_KUH-X%23% As X%23%
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4-X%31%-X%28%-X%29%-X%02%-X%04%14()
  BOX_1:
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

## [図179]

定義規則35で定義されるベクトルS4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
  Private W%X30%-X%31%-X%28%-X%29%-IN_Conditions_KEY As Integer
Rem PUB4N4
  Private W%X30%-X%31%-X%28%-X%29%-IN_ACCESS_KEY As Integer
Rem PUB4N4
  Private CTR-X%31%-X%28%-X%29%-04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
  Private CNS_NOT_KUH_Integer As Integer
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4-X%31%-X%28%-X%29%-X%02%-X%04%18()
  BOX_1:
Rem PUB04STS
  If CTR-X%31%-X%28%-X%29%-04_END <> True Then GoTo BOX_E
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

## [図180]

定義規則37で定義されるベクトルS4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB4N4
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04015()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図181]

定義規則38で定義されるベクトルS4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W%300_0%310_0%280_0%290_OUT_Conditions_KEY As Integer
Rem PUB4N4
Private W%300_0%310_0%280_0%290_OUT_ACCESS_KEY As Integer
Rem PUB4N4
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_Integer As Integer
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図182]

定義規則 4 ( ) で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04016()
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図183]

定義規則 4 1 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04016()
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図184]

定義規則 4 2 で定義されるベクトル S 4

```

Option Explicit
Rem #DEF PUB
Rem PUB4N4
Private WE%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEF PRV
Rem PRV7R4
Private CTRL_WE%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%020(0%04016())
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX E
BOX 2:
Rem PUBKK
BOX 3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図185]

定義規則 4 3 で定義されるベクトル S 4

```

Option Explicit
Rem #DEF PUB
Rem PUB4N4
Private WE%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEF PRV
Rem PRV7R4
Private CTRL_WE%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%020(0%04016())
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図186]

定義規則 4 4 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W@%30@_@%31@_@%28@_@%29@_@%02@ As @%23@
Rem PUB4N2
Private CTR_@%31@_@%28@_@%29@_04_END As Boolean
Rem #DEFEND
Rem #DRPPRV
Rem PRV7R4
Private CTRL_W@%30@_@%29@_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_@%23@ As @%23@
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%31@_@%28@_@%29@_@%02@@%04@16()
BOX_1:
Rem PUB04STS
If CTR_@%31@_@%28@_@%29@_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図187]

定義規則 4 5 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W@%30@_@%31@_@%28@_@%29@_@%02@ As @%23@
Rem PUB4N2
Private CTR_@%31@_@%28@_@%29@_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W@%30@_@%29@_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_@%23@ As @%23@
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%31@_@%28@_@%29@_@%02@@%04@16()
BOX_1:
Rem PUB04STS
If CTR_@%31@_@%28@_@%29@_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図188]

定義規則 4 6 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W%X300_@%310_@%280_@%290_@%020(@%260,@%270) As @%230
Rem PUB4N2
Private CTR_@%310_@%280_@%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W%X300_@%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_@%230 As @%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%310_@%280_@%290_@%020@%04@16()
BOX_1:
Rem PUB04STS
If CTR_@%310_@%280_@%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図189]

定義規則 4 7 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W%X300_@%310_@%280_@%290_@%020(@%260,@%270) As @%230
Rem PUB4N2
Private CTR_@%310_@%280_@%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W%X300_@%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_@%230 As @%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%310_@%280_@%290_@%020@%04@16()
BOX_1:
Rem PUB04STS
If CTR_@%310_@%280_@%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```



## [図190]

定義規則 48 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
  Private W%X300_0%X310_0%X280_0%X290_0%X020 As 0%X230
Rem PUB4N2
  Private CTR_0%X310_0%X280_0%X290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
  Private CTRL_W%X300_0%X290_ING_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X02004016()
  BOX_1:
Rem PUB04STS
    If CTR_0%X310_0%X280_0%X290_04_END <> True Then GoTo BOX_E
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

## [図191]

定義規則 49 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
  Private W%X300_0%X310_0%X280_0%X290_0%X020 As 0%X230
Rem PUB4N2
  Private CTR_0%X310_0%X280_0%X290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
  Private CTRL_W%X300_0%X290_ING_FLG As Boolean
Rem PUBKU
  Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X02004016()
  BOX_1:
Rem PUB04STS
    If CTR_0%X310_0%X280_0%X290_04_END <> True Then GoTo BOX_E
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

## [図192]

定義規則 50 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04016()
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図193]

定義規則 51 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem PUB4N2
Private CTR_0%310_0%280_0%290_04_END As Boolean
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W0%300_0%290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04016()
BOX_1:
Rem PUB04STS
If CTR_0%310_0%280_0%290_04_END <> True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図194]

定義規則 5 2 で定義されるベクトル S 4

```

Option Explicit
Rem *** VECDEF
Rem #DEFMUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBPKY04
Private CTRL_W0%300_0%310_0%280_0%290_0%020_OUT_Conditions_KEY_NOT_VALID_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04017()
BOX 1:
Rem PUBPKY04
If CTRL W0%300 0%310 0%280 0%290 OUT Conditions KEY NOT VALID FLG = True Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図195]

定義規則 5 3 で定義されるベクトル S 4

```

Option Explicit
Rem *** VECDEF
Rem #DEFMUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEFMUB
Rem PUBPKY04
Private CTRL_W0%300_0%310_0%280_0%290_0%020_OUT_Conditions_KEY_NOT_VALID_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04017()
BOX 1:
Rem PUBPKY04
If CTRL W0%300 0%310 0%280 0%290 OUT Conditions_KEY_NOT_VALID_FLG = True
Then GoTo BOX_E
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図196]

定義規則 58 で定義されるベクトル S 4

```

Option Explicit
Rem #DEF PUB
Rem PUB4M4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEF PRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図197]

定義規則 59 で定義されるベクトル S 4

```

Option Explicit
Rem #DEF PUB
Rem PUB4M4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEF PRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図198]

定義規則 6 0 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図199]

定義規則 6 1 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04019()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図200]

定義規則6 2 で定義されるベクトルS 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図201]

定義規則6 3 で定義されるベクトルS 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020 As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

[図202]

定義規則 6 4 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4M4
Private W%X300_0%X310_0%X280_0%X290_0%X020(0%X260,0%X270) As 0%X230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

[図203]

定義規則 6 5 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4M4
Private W%X300_0%X310_0%X280_0%X290_0%X020(0%X270,0%X270) As 0%X230
Rem #DEFEND
Rem #DEFPRV
Rem PRV7R4
Private CTRL_W%X300_0%X290_ING_FLG As Boolean
Rem PUBKU
Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```



[[図204]]

定義規則 6 6 で定義されるベクトル S 4

```

Option Explicit
Rem hDEFPUR
Rem PUB4N4
Private W%X30E_@%31E_@%28E_@%29E_@%02E As @%23E
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_@%23E As @%23E
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%31E_@%28E_@%29E_@%02E@%04E18()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

[[図205]]

定義規則 6 7 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUR
Rem PUB4N4
Private W%X30E_@%31E_@%28E_@%29E_@%02E As @%23E
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_@%23E As @%23E
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_@%31E_@%28E_@%29E_@%02E@%04E18()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図206]

定義規則 6 8 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCFND

```

## [図207]

定義規則 6 9 で定義されるベクトル S 4

```

Option Explicit
Rem #DEFPUB
Rem PUB4N4
Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%0200%04018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCFND

```

## [図208]

定義規則 7 0 で定義されるベクトル S 4

```

Option Explicit
Rem *** VECDEF
Rem #DEFPUB
Rem PUB4N4
Private W%X300_0%X310_0%X280_0%X290_0%X020 As 0%X230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X02000004018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図209]

定義規則 7 1 で定義されるベクトル S 4

```

Option Explicit
Rem *** VECDEF
Rem #DEFPUB
Rem PUB4N1
Private W%X300_0%X310_0%X280_0%X290_0%X020 As 0%X230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
Private CNS_NOT_KUH_0%X230 As 0%X230
Rem #DEFEND
Public Sub Main()
Rem #VECREP Private Sub S4_0%X310_0%X280_0%X290_0%X02000004018()
BOX_1:
BOX_2:
Rem PUBKK
BOX_3:
BOX_4:
BOX_5:
BOX_6:
BOX_7:
BOX_E:
End Sub
Rem *PRCEND

```

## [図210]

定義規則 7 2 で定義されるベクトル S 4

```

Option Explicit
Rem *** VECDEF
Rem #DEFPUB
Rem PUB4N4
  Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
  Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%020(0%04018())
  BOX_1:
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

## [図211]

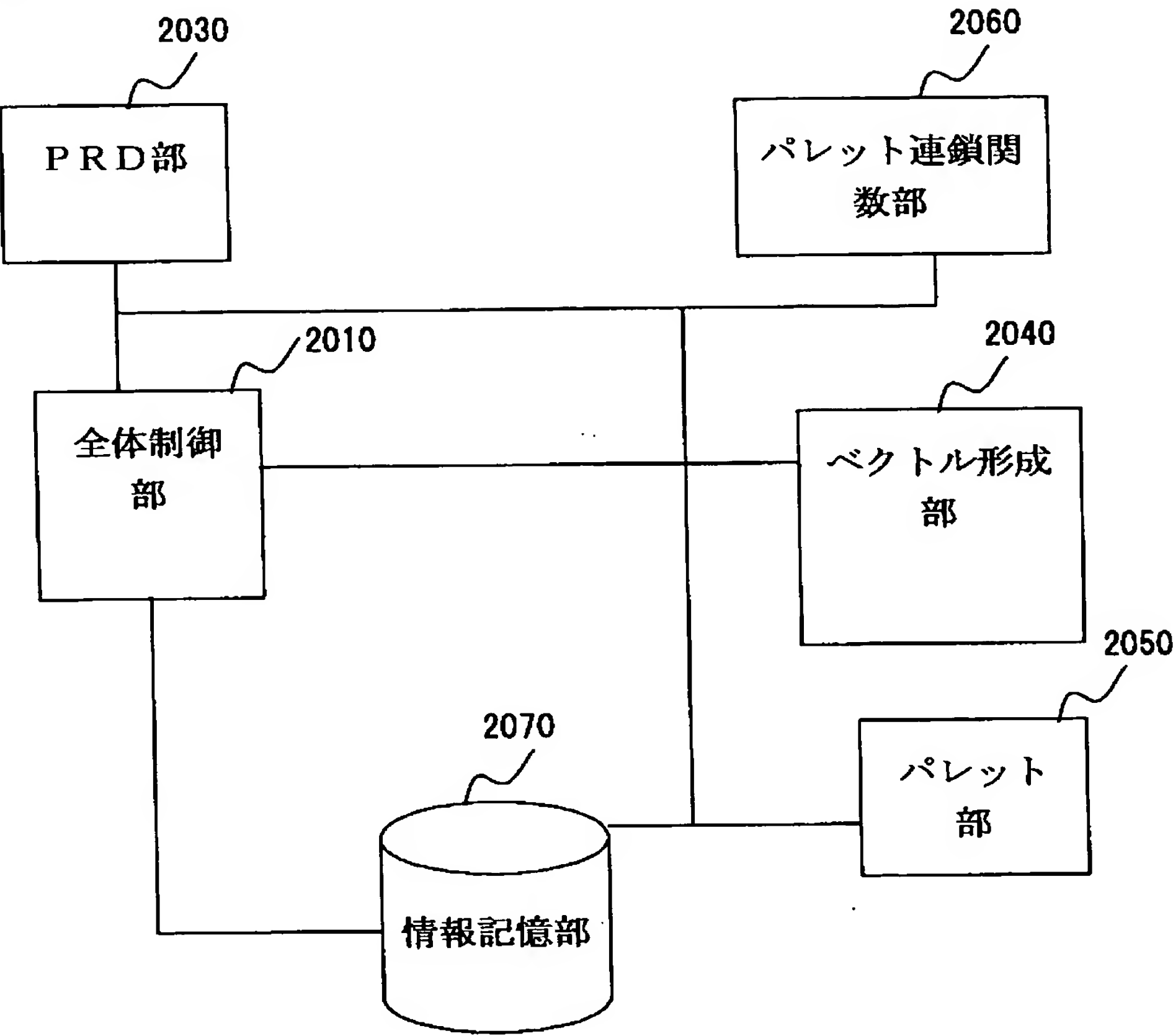
定義規則 7 3 で定義されるベクトル S 4

```

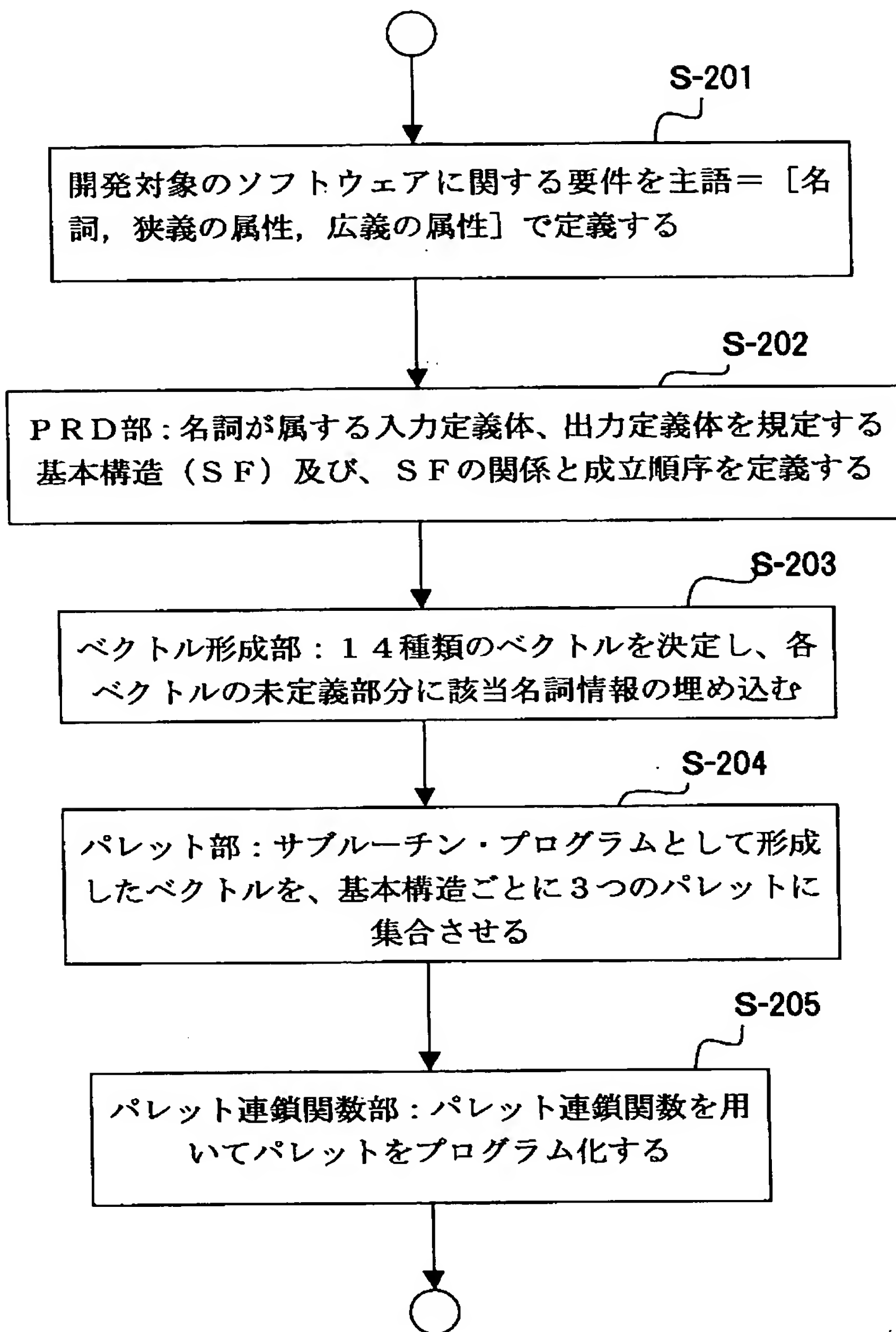
Option Explicit
Rem *** VECDEF
Rem #DEFPUB
Rem PUB4N4
  Private W0%300_0%310_0%280_0%290_0%020(0%260,0%270) As 0%230
Rem #DEFEND
Rem #DEFPRV
Rem PUBKU
  Private CNS_NOT_KUH_0%230 As 0%230
Rem #DEFEND
  Public Sub Main()
Rem #VECREP Private Sub S4_0%310_0%280_0%290_0%020(0%04018())
  BOX_1:
  BOX_2:
Rem PUBKK
  BOX_3:
  BOX_4:
  BOX_5:
  BOX_6:
  BOX_7:
  BOX_E:
End Sub
Rem *PRCEND

```

[図212]



[図213]



# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/013806

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl<sup>7</sup> G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl<sup>7</sup> G06F9/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2004
Kokai Jitsuyo Shinan Koho	1971-2004	Toroku Jitsuyo Shinan Koho	1994-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2001-5651 A (The Institute of Computer Based Software Methodology and Technology), 12 January, 2001 (12.01.01), Full text; all drawings & WO 2000/079385 A1 & CA 2414110 A & EP 1244006 A1	1
A	WO 2001/035213 A1 (Information System Development Institute), 17 May, 2001 (17.05.01), Full text; all drawings & EP 1248189 A1 & CA 2359079 A & AU 1305401 A & NZ 512711 A	1

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance  
 "E" earlier application or patent but published on or after the international filing date  
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)  
 "O" document referring to an oral disclosure, use, exhibition or other means  
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention  
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone  
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art  
 "&" document member of the same patent family

Date of the actual completion of the international search  
14 December, 2004 (14.12.04)

Date of mailing of the international search report  
11 January, 2005 (11.01.05)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.



## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> G06F9/44

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> G06F9/44

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2004年
日本国実用新案登録公報	1996-2004年
日本国登録実用新案公報	1994-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 2001-5651 A (ソフトウェア生産技術研究所) 2001.01.12, 全文, 全図 & WO 2000/079385 A1 & CA 2414110 A & EP 1244006 A1	1
A	WO 2001/035213 A1 (株式会社アイエスデー研究 所) 2001.05.17, 全文, 全図 & EP 1248189 A1 & CA 2359079 A & AU 1305401 A & NZ 512711 A	1

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの  
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの  
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)  
「O」 口頭による開示、使用、展示等に言及する文献  
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの  
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの  
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの  
「&」 同一パテントファミリー文献

国際調査を完了した日

14.12.2004

国際調査報告の発送日

11.1.2005

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)  
郵便番号100-8915  
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

漆原 孝治

5B

9366

電話番号 03-3581-1101 内線 3546

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**